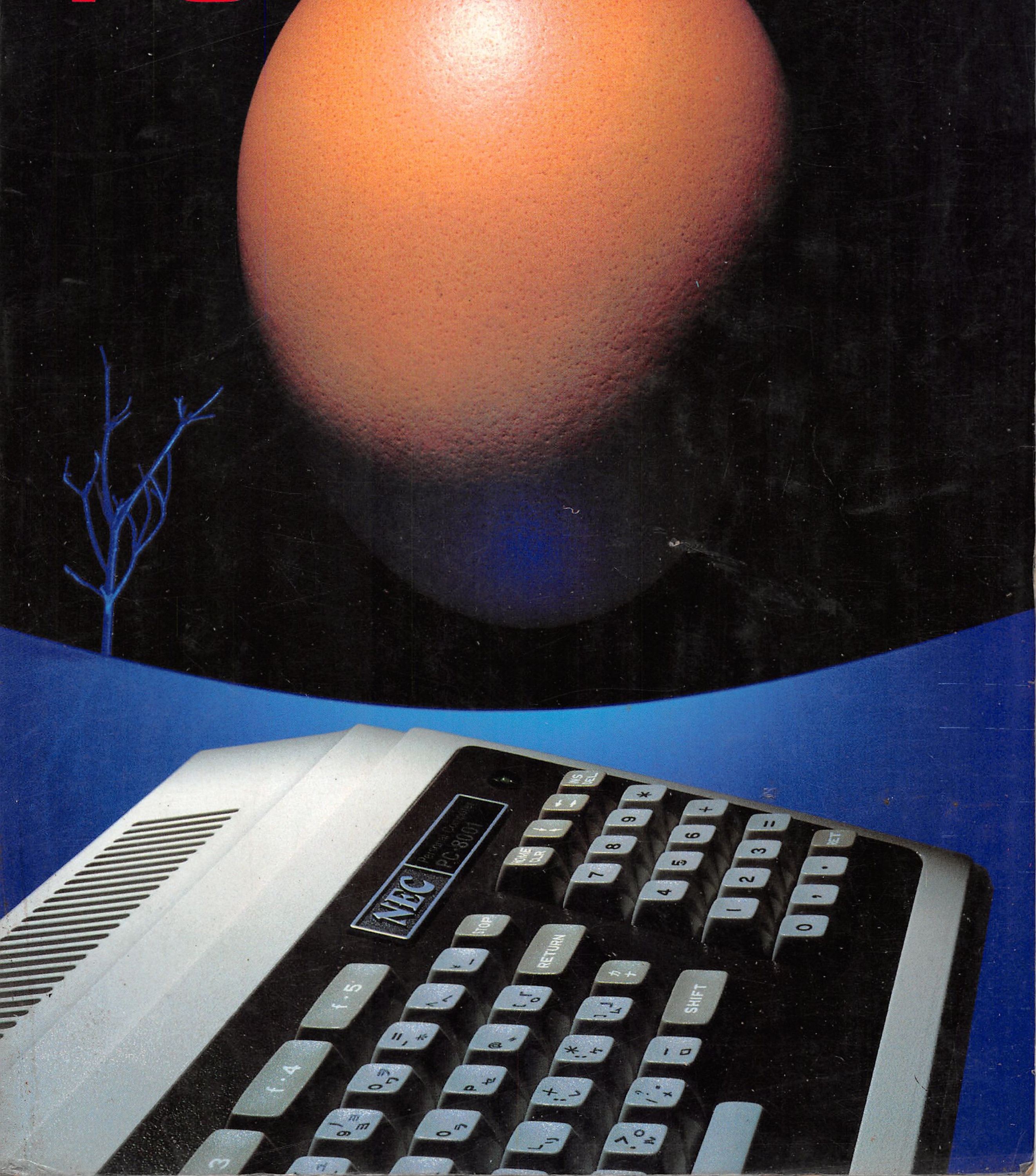
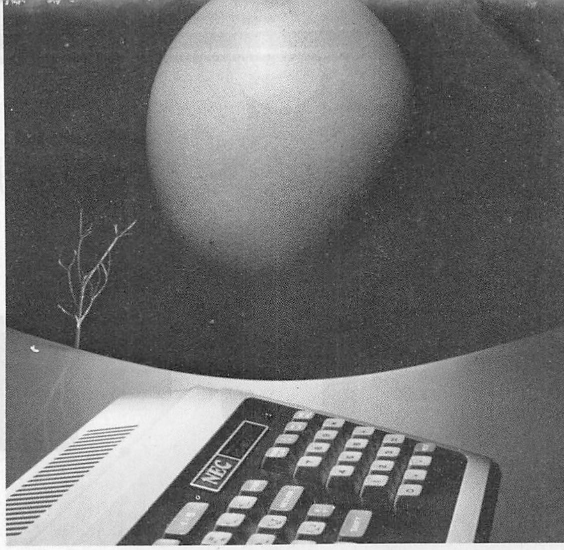
I/O 別冊

PC-8001活用研究



I/O 別冊

PC-8001活用研究



CONTENTS

11日間で パーソナル・コンピュータをマスターする！

PC-8001入門

■名古屋市工業研究所　河合勝司

まえがき

小学生から大人まで，我が国もパーソナル・コンピュータの愛好家や利用者が増加してきました．そして大勢の人がマイコン教室や講習会に参加し，勉強しています．また，マイコンに関する本が特別コーナーにずらりと並んでいる書店も増えてきました．そして，家電機器販売店にもマイコンが並び，いよいよパーソナル・コンピュータ時代の到来を物語っています．

筆者は昭和52年から名古屋の朝日文化センターでマイコンの講座を担当し，多くの初心者の方々と接しています．初心者がプログラムを独力で作ったり，理解できるようになることは，大変，困難なことですが，勉強のしかたさえ間違えなければ，自動車の運転免許証を取得するより容易だと話しています．

そこで，ここでは基礎的事項を徹底的に実験してみることに重点をおいて書きました．文法や，命令，コマンドなどの詳細な説明はメーカーのマニュアルを参照することにし，一切，説明していません．ここに出てくる実験をし，理解した時点でメーカーのマニュアルの該当する事項を読んでください．きっと，書いてあることが理解できるようになると思います．

本書が起爆剤となって，１人でも多くのパーソナル・コンピュータのファンが増え，パーソナル・コンピュータを駆使して，より豊かな生活をおくる人々の多からんことを祈ります．

1 第1日 キーボードと画面の操作

〔目標１〕
数字，英字，カナ文字，特殊記号などの文字キーの押し方，ファンクション・キー，画面制御用キーの機能と操作方法を理解します．
〔目標２〕
ＣＲＴディスプレイの行と桁の設定法，カーソルやスクロールの操作方法，キャラクタ・モードとグラフィック・モードの制御法，カラー制御の方法について理解します．

マイコンの操作法

マイコンを使うためにはプログラムを作らなければなりません．プログラムはキーボードを使ってマイコンのメモリの中に格納します．

図１-１　キーの配列

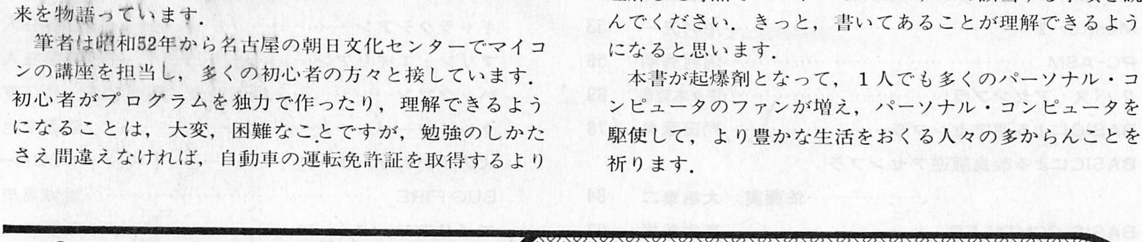


図1-2　グラフィック・シンボルのキー配置

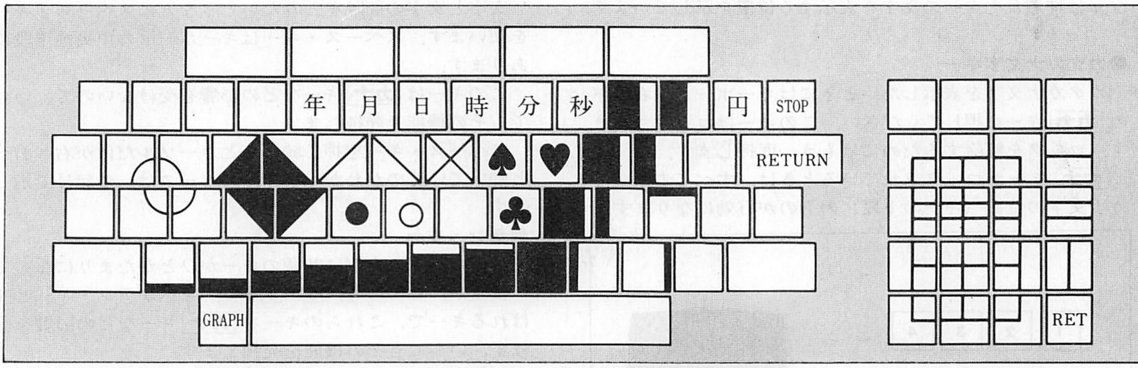
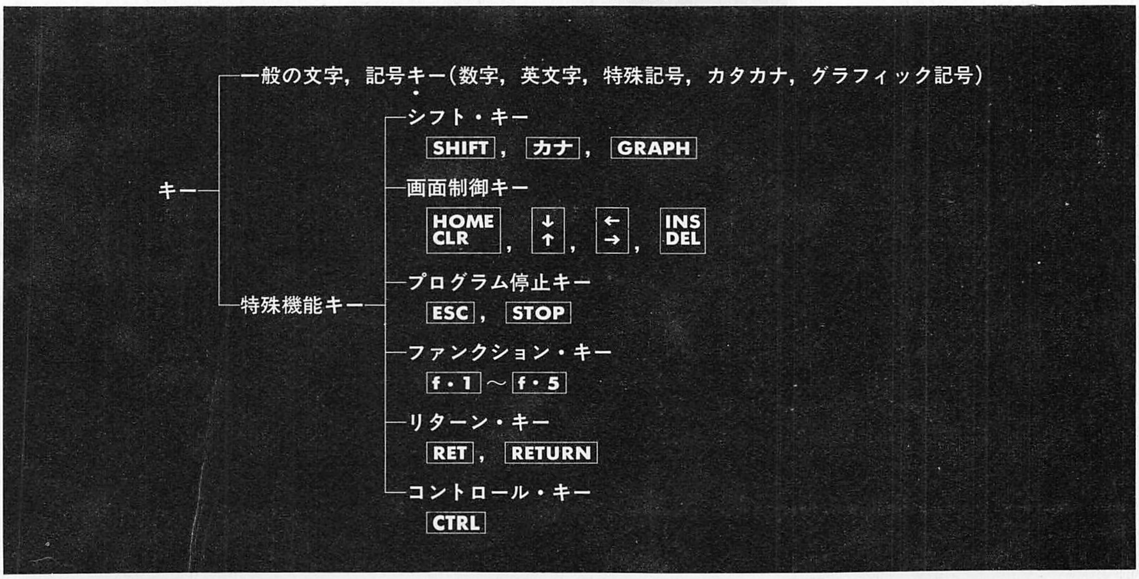


図1-3　キーの分類図



マイコンに接続してあるキーボードにはいろいろなキー配例のものがありますが，PC-8001では図1-1のようなキーの配例になっています．また，マイコンにはグラフィックが楽しめるように図1-2のようなグラフィック用のキー機能を兼用できるものもあります．

第1日目はこれらのキー操作を実際にマイコンのキーにさわりながら実験的に理解し，キーボードに慣れ，親しむことを学習の目標とします．

1 キーボードの操作

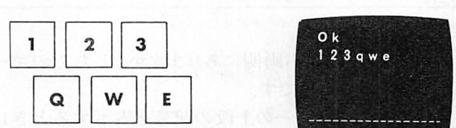
マイコンのキーボードにはたくさんのキーが配置されています．数字，英文字，特殊記号などの一般的なキーと，その他の特殊機能キーに大別できます．

これから，キーの操作法について学ぶわけですが，図1-3にキーの分類を示します．

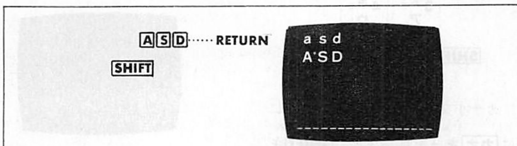
●数字と英文字キー

PC-8001に電源を投入すると，ディスプレイ上にいろいろなメッセージが表示され，最後にOkと表示されてキー入力待ちの状態になります．

数字キーや英文字キーを適当に押すと，押されたキーの文字がディスプレイに表示されます．このとき，英文字は小文字で表示されます．もし，大文字を表示したければ，SHIFT キーを押したまま英文字キーを押してください．



上のキーを押すと右画面のように数字，英小文字が表示される．



❶A，S，Dのキーを押すと，asdの小文字が表示される．
❷RETURNキーを押すと改行され，カーソルが左端に位置する．
❸SHIFTキーを押したままでA，S，Dのキーを押すと英文字のASDが表示される．

〔注意〕

1．数字や英文字キーを長く押し続けると同じ文字が連続的に表示されるので，1文字だけ表示させたいときは〝タッチ〟打法でキーを押してください．

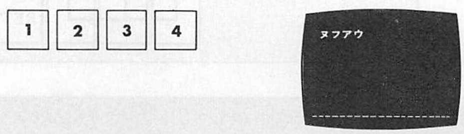
2．Okという文字の下でついたり消えたりしている（ブリ

ンクしている）四角の記号ををカーソルといいます．カーソルは押されたキーの文字を表示する位置を示しています．

●カタカナ文字キー

カタカナ文字を表示したいときにはキーボードの右端下の カナ キーを押してください．このキーはロックされます．ロックを解除するためにはもう一度押します．

カナ キーがロックされているときは，すべてのキーはカナ文字の位置（キーの下段）のものが有効になります．

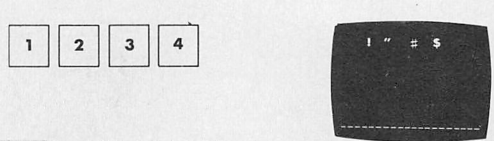


カナキーをロックして(押し込んだ状態)，上のキーを順番に押すと**右画面**のように表示されます．

カナ文字以外のときは，ロックを解除（もう一度押してキーを上げて）してキーを押してください．

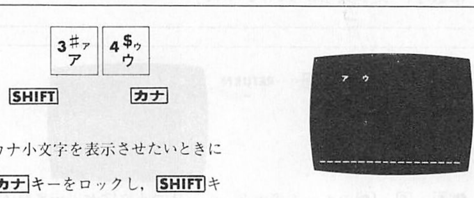
●シフト・キー

SHIFT キーは英大文字を表示するときに使いましたが，特殊記号やカタカナの小文字を表示するのにも使います．



SHIFTキーを押したまま上のキーを押すと，**右図**のように表示されます

〔注意〕
1. SHIFT キーは左右両側にありますが，どちらのキーを押しても機能は同じです．
2. SHIFT キーは各キーの上段の記号を表示するときに使われます．やってみてください．



カナ小文字を表示させたいときにはカナキーをロックし，SHIFTキーを押したまま上のキーを押すと，**右図**のように表示されます．

まとめ 文字，特殊記号キー

表示したい文字，記号	キー操作
3	3 #ァ ア
#	SHIFT 3 #ァ ア
ア	カナ 3 #ァ ア
ァ	カナ SHIFT 3 #ァ ア

●スペース・キー

文字と文字の間隔をとりたいときなどにスペース・キーを使います．スペース・キーはキーボードの中央最下段にあります．

このキーは カナ キーなどの影響を受けないので，いつでもその機能を発揮します．

スペース・キーを押し続けるとカーソルだけが右へ右へと動いていくのがわかります．スペースは␣の記号で示します．

●テン・キー

キーボードの右側に20個のキーがひとかたまりになっていますが，このうち，0～9までのキーがテン・キーと呼ばれるキーで，これらのキーも カナ キーなどの影響を受けず，いつでもその機能を発揮します．

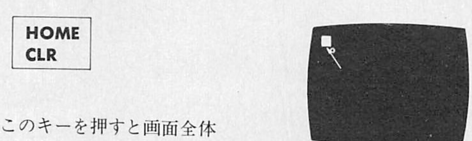
数字入力の場合はこのキーを使うと便利でしょう．右下すみにある RET キーは RETURN キーと機能が同じです．どちらを使ってもかまいません．

この20個のキーにある＊，＋，＝，，，・の各特殊記号キーについても カナ キーのロック状態に影響されずに使えます．

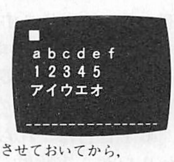
●画面制御用キー

さて，ディスプレイの画面にはいろいろな文字が表示されていると思いますが，黒板の字などを消すようにディスプレイの画面を消すには HOME CLR という，テン・キーの式上に位置するキーを使います．

このキーを押すと，画面全体が消去され，カーソルが画面の左上隅(すみ)に移動します．



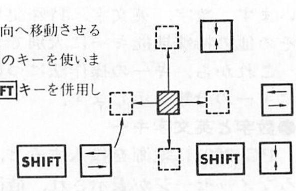
このキーを押すと画面全体が消去され，カーソルが左上隅に移動します．



カーソルだけを画面の左上隅に移動するにはSHIFTキーとHOME CLRキーを用います．

この左上隅を**ホーム・ポジション**と呼びます．

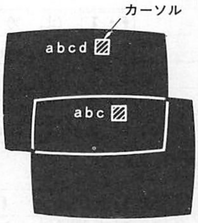
この実験は適当に画面にいくつかの文字を表示させておいてから，実行して確かめてください．



カーソルを右や左，上や下の方向へ移動させるにはテン・キーの上側にある⇕ ⇆のキーを使います．上段で示される方向にはSHIFTキーを併用します．

〔注意〕
1. 同一方向に数行とか数桁分だけ移動させたいときはキーを押し続けてください．
2. カーソルが最上行の左はしとか最下行の右はしとか画面の緑にあるときには，移動状態が変わるので注意してください．

キー入力した文字を取り消したいときには INS/DEL キーを押せば，直前にキー入力した文字を消去できます．すなわち，カーソルの左1桁が消去され，カーソルはその消去された桁の位置に移動します．右図で d を消去するためには INS/DEL を押せばよいことになります．



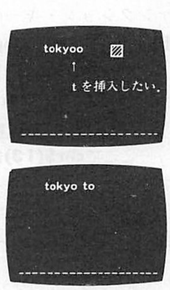
カーソルの左数桁を消去したいときは INS/DEL キーを消したい数だけ押します．やってみてください

キー入力の終ったあとで，脱字とか，編集上，文字などを追加・挿入したいときがあります．

そのときには SHIFT キーと INS/DEL キーを併用で2文字の挿入をするための空白部分を作ります．

右図の場合，まずカーソルを1桁左へ移動し，SHIFT，INS/DEL を押すとスペースが1個入ります．

カーソルを SHIFT ⇆ でスペース上にもってきて，T をキーインします．



これらの作業はプログラムの修正時によく行ないますが，この作業が終ったら必らず RETURN キーを押してカーソルを左端へ移動してください．

後でプログラムの編集，デバックについては説明しますが，これは誰もがよく忘れることです．

複数個の文字を挿入するにはどうしたらよいでしょうか．1度，実際に実験してみてください．

まとめ 画面制御キー

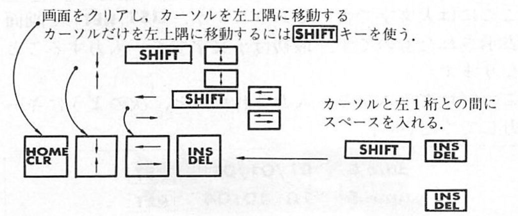


図1-4

HOME/CLR	画面をクリアし，カーソルを左上すみに移動する．
SHIFT HOME/CLR	カーソルだけを左上すみに移動する．
↕	カーソルを上の行に移動する．
SHIFT ↕	カーソルを下の行に移動する．
⇆	カーソルを右に1桁だけ移動する．
SHIFT ⇆	カーソルを左に1桁だけ移動する．
INS/DEL	カーソルの左1桁を消去する．
SHIFT INS/DEL	カーソルと左1桁との間にスペースを入れ，右側を移動する．

プログラム修正用として下記のようなキーもあります．

CTRL Eｨ	ERASE	カーソルより後ろにある，その行の残りの行をすべて消去する．
CTRL Nﾐ	NEXT WORD	空白や特殊記号に続く英文字にカーソルを右方向へ移動する．
CTRL Bｺ	BACK WORD	空白や特殊記号に続く英文字にカーソルを左方向へ移動する．
CTRL Jﾏ	JUMP WORD	カーソル以降のその行の残りの部分を次の行にずらす(マルチステートメント用)．

●グラフィック・キー

図1-2にはPC-8001で入力できるグラフィック記号について示しました．これらの記号を入力するにはスペース・キーの左隣りにある GRAPH キーを押したまま希望のグラフィック用キーを押すことになります．

GRAPH キーを押したまま 5 6 7 のキーを順番に押していくと右画面のように年月日というグラフィック文字が表示されます．



〔演習〕

①画面に時・分・秒・10,000円などと表示してみてください．

②ハート，スペード，ダイヤ，クローバーの記号を画面に表示してみてください．

①グラフィック用キーを使って，作表することを確かめてみてください．

②**width, 25** RETURN としてから，同様な図を作ってみてください．

③作表用にはテンキー側のキーを使います．

●プログラム停止用キー

実行中のプログラムを停止させたり，一時中断させたりしたい場合が生じたときには STOP キーや ESC キーなどを使います．

❶画面を消去してください…… HOME/CLR を押す．

❷次のようにキー入力してください．

（簡単な印刷のプログラムをマイコンに格納します）．

プログラム1-1

```
10␣  x = 0      RET
20␣  x = x + 1  RET
30␣  print  x   RET
40␣  goto 20    RET
```

❸キー操作を間違えた場合は，前述した通りのやりかたで直ししてください（カーソルの移動と文字の消去，挿入）．

❹**list** RET とキー入力してみてください．
このとき画面に❷の通りのプログラム・リストが表示されます．もう一度，間違いがないか確認してください（**list** で表示されたものは大文字になっています）．

❺間違いがあれば再度入力してください．

❻間違いがなければ，**run** RET とキー入力してください．キー入力が終りしだい，画面に1から順番に数値が表示されます．

❼数値が繰り返し表示されていますが，ここで ESC キーを押すと画面の動きが中断されます．このあと，任意のキーを押すとまた再開され，引き続き画面が動きます．



❽ STOP キーを押すとピーと鳴ってBreak in ××を表示し，プログラムは完全に停止します．
ここで××は20か30か40のいずれかです．

❾画面の最下端に数値が表示されるたびに1行分上に押し上げられ，画面の最上行の数値が見えなくなります．この動きを**スクロール**と呼んでいます．

●ファンクション・キー

runとか**print**とか**goto**などという命令はよく使います．そのつど，1文字ずつキー入力をしてもよいのですが，G O T O と4回もキー入力するのは面倒なので，これを特殊なキーに代行させようとするものがファンクション・キーです．

キーボードの最上段にある5個の大きなキーがファンクション・キーです．真中の f・3 のキーを押すと**goto**を入力したのと同じことになります．

ここで，␣キーはスペース・キーを意味します．

❶ f・1 キーはカーソルを右へ8桁ずつ移動させるときに使います．

❷ f・2 キーは**プログラム1-1**で作ったプログラムのうち，行番号といわれる10とか20とか30などのものを自動的に発生させるときに使います．
この f・2 キーはautoが定義されています．auto機能を解除したいとき（行番号を自動的に表示させるのをやめたいとき）は STOP キーを押してください．

❸gotoは**プログラム1-1**のプログラムでも使いましたが，f・3 のキーを1回押すだけで画面にgotoが表示できます．

❹どのようなプログラムがメモリに格納されているのか，そのリストを画面に表示するときに使うのが f・4 のキーです．
このキーはlistが定義されているので f・4 と RET を押すことによって画面にプログラムのリストを表示させることができます．

❺プログラムを実行させるために**プログラム1-1**では r u n と3回キー入力をして RET キーを押しましたが，かわりに，f・5 キーを1回押せばプログラムを実行することができます．
どのキーにどんな機能が定義されているかは，ディスプレイの最下行に表示されているので確かめてください．

❻**print**などは SHIFT キーを押すとディスプレイの最下行の中央に表示されているので SHIFT キーと f・3 を押せばprintと画面に表示できます．

まとめ ファンクション・キー

図1-5

f・1	は，カーソル移動（8桁単位）
f・2	は，auto
f・3	は，goto
f・4	は，list
f・5	は，run RET

SHIFT を押したまま

f・1 を押すと	time$	⇨ f・6	とする
f・2 を押すと	key	⇨ f・7	とする
f・3 を押すと	print	⇨ f・8	とする
f・4 を押すと	list. RET	⇨ f・9	とする
f・5 を押すと	cont RET	⇨ f・10	とする

❶各ファンクション・キーには以上の定義以外にそれぞれ自由に最大15文字まで定義することができます．
定義されている内容の文字表示は6文字または12文字表示されます．

❷ファンクション・キーの定義のしかた．
（例1） f・1 キーにinputを定義するには，
key1, "input" RET とします．
（例2） f・7 キーに**end** RET を定義するにはkey7,"end" + chr$(13) RETURN とします．
ここで**chr$(13)**は RETURN コードを意味します．

❸各ファンクション・キーが何に定義されているかを確かめるには，画面の最下行のリスト表示をみることのほかに，**key list** RET で画面に f・1 ～ f・10 までを表示することができます．



new RET とキー入力してから次のプログラムをキー入力して実行してみてください．**new**コマンドはメモリに入っているすべてのプログラムを消去します．

プログラム1-2

```
10  LOCATE  31, 2: PRINT  DATE$
20  LOCATE  31, 4: PRINT  TIME$
30  GOTO  20
```

ここには大文字で書いてありますが，**LIST** RET で画面に表示されたものです．最初は小文字でキー入力することになります．

このプログラムがキー入力されたあと，次のようにキー入力してください．

date $="81/01/01" RET
time $="10:30:04" RET

これでマイコンに内蔵されているクロックに日付けと時刻がセットされました．

HOME CLR キーを押して画面を消去し，次に，f･5 キーを押してプログラムを実行してみてください．

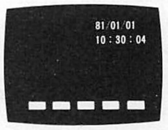
★時刻が変っていくことを確かめてください．

★現在時刻にセットし直して，再びプログラムを実行してみましょう．

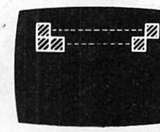
❶STOP キーを押す

❷time$＝"××：××：××" RET に現在の時，分，秒を入力する．

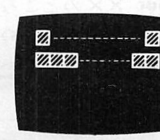
❸RUN RET または f･5 を押す．



❶プログラムの入力が終ったら，width, 25 RET とキー入力した後 f･5 のキーを押してプログラムを走らせてください．



❷ STOP キーを押してプログラムを停止させ，HOME CLR キーを押し，画面をクリアして width, 20 RET として f･5 を押してください．



❸ 1行目を2行目の間隔を比較してみましょう．

2 画面の操作

マイコンではディスプレイを使ってプログラムを作ったり，実行結果を表示したりします．この場合，文字の大きさや，表示の質が問題になります．

1行に表示できる桁数は普通は40文字ですが，PC-8001では40文字/行のほか，36文字/行，72文字/行，80文字/行にしたりすることができます．

同様に，行数も20行/画面と25行/画面のいずれかを選択することができます．

これらの行数や1行の文字数を指示することがいつでもできるので，プログラムの中でも画面を操作することができます．

その他，グラフィック用としては画面が，横160ドット，縦100ドットに分解されて，ドット表示になります．

このように，キャラクタ（文字）モードとグラフィック・モードの2つがあります．

その他，白黒モードとカラーモードの切り換えもできます．ここでは，これらの画面の操作法について学習します．

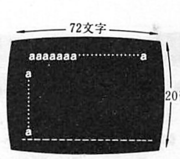
●画面の文字数/行，行数/画面の指定

1行に表示できる文字数は通常は40字に設定されています．また，行数は20行になっています．

もし，1行あたりに80字を表示し，1画面あたりの行数を25行にしたいときは，**width80, 25 RET** とすれば，1行が80字，25行の画面に設定されます．

1行を72文字，20行/画面を設定するには，**width 72, 20 RET** とする．

任意の文字キーを押し続けて表示される文字数を確かめてください．



new RET をキー入力してから次のプログラムを入力してください．

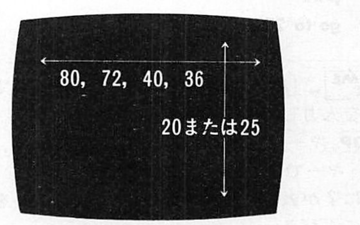
プログラム1-3

```
10 Print "■";          ■は GRAPH  , キーを
20 Goto 10                    押します．
```

まとめ widthを使った画面操作

width 〈文字数/行〉，〈行数/画面〉

80	**,**	**25**
80	**,**	**20**
72	**,**	**25**
72	**,**	**20**
40	**,**	**25**
40	**,**	**20**
36	**,**	**25**
36	**,**	**20**



●画面モード（カラー制御）

ディスプレイの使用法として白黒で使うのか，カラーで使うのか，または文字だけを表示するのか，グラフを描かせるのかによって，いろいろな使い方があります．

これらの画面モードを決めるためにCONSOLEや，COLORのステートメントがあります．

❶console , , , 0 RET とキー入力して任意のキーを押し続けてください．



❷console , , , 1 RET とキー入力して任意のキーを押し続けてください．



色は白

❸次に**color 2**とキー入力してください.
　Okと赤色で表示されます. 任意のキーを押し続けて赤色で文字が表示されることを確かめてください.



color ××のところを3, 4, 5, 6, 7などに変更してどんな色になるか確かめてください.

まとめ　白黒モードとカラーモード

console ,,,0　白黒モード
console ,,,1　カラーモード
色は**color ×**で指定

0	黒
1	青
2	赤
3	マゼンタ
4	緑
5	シアン
6	黄
7	白

new **RET** をキー入力し, 次のプログラムを入力してカラーモードに慣れましょう.

プログラム1-4

```
10 console ,,,1
20 input "1~7マデノカズノウチ ドレカオ イレヨ"; x
30 color x
40 print "カラーディスプレイ ノ ジッケン"
50 print
60 go to 20
```

❶ **HOME CLR**, **AUTO**, **RET** キーなどを使って上のプログラムを入力してください.
❷ **STOP** キーでプログラム入力を終了します.
❸ **f･5** キーでプログラムを実行します.
❹画面に？が表示されたら**1**~**7**の数字キーをどれか1つ押してください.
❺いろいろな数字キー(**1**~**7**)を押して色を確かめてください.

●キャラクタ・モードとグラフィック・モード

画面にグラフや絵を描くには1画面の密度を高くした方がきれいな絵を描くことができます. グラフィック・モードでは横がキャラクタ・モードのときの2倍, 縦が4倍の密度になります.

最高の1画面表示文字数は80×25文字だから, 160ドット×100ドットの密度になります. 後の方で, グラフィックスについて詳細な説明をするので, ここではキャラクタ・モードとグラフィック・モードの画面の切り換え法についてだけ説明します.

グラフィック・モードにするには**color**ステートメントの第3番目のパラメータで指定します.

グラフィック・モードのとき………**color　0, 0, 1**
キャラクタ・モードのとき…………**color　0, 0, 0**

なにも指定しないときはキャラクタ・モードになっています.

●画面のスクロール

通常の画面は最下行まで表示されると次に最上行が消えてしまい, 1行ずつ上へ移動しますが, このような画面の動きを**スクロール**といいます.

スクロールする行の範囲を制限したいときなど消えてしまう行を制御したいことがあります. そのためには, **console**ステートメントの第1パラメータと第2パラメータで指示します.

new **RET** とキー入力してから, 次のプログラムをを入力してください.

プログラム1-5

```
10 console 5,10,0,0
20 x=0
30 x=x+1
40 print x
50 goto 30
```



上のプログラムの入力が終わたら **f･5** のキーを押して実行してみてください.
スクロールの開始行は5行目
スクロールの範囲は5行目から10行分であることがわかります.

consoleステートメントを使ってスクロールの範囲, 開始行をいろいろ変えて実験してみてください.

●ファンクション・キーの表示

画面にファンクション・キーの表示がいつもあるのは目ざわりであるとか, 応用プログラムなどのときには使わないとかの事情で画面上からファンクション・キーの表示を消去したいときがあります. そのときには, **console**ステートメントの第3パラメータによって制御することになっています.

console　0, 20, 0, 0……ファンクション・コード消去
console　0, 20, 1, 0……ファンクション・コード表示

まとめ　画面モード

白黒モード **console,,,0**
- 白黒でキャラクタ・モード　**color 0, 0, 0**
- 白黒でグラフィックモード　**color 0, 0, 1**

カラーモード **console,,,1**
- カラーでキャラクタ・モード　**color 2, 0, 0**
- カラーでグラフィック・モード　**color 6, 0, 1**

（6＝色指定）

まとめ　colorについて

❶カラーモードのとき(**console,,,1のとき**)にのみ色指定が有効となる.
color〈カラー〉, 〈nullキャラクタ〉, 〈グラフィック・スイッチ〉

クロ	0	画面クリア時の表示文字	グラフィック・モード 1
アオ	1	通常はスペース	キャラクタ・モード 0
アカ	2		
マゼンダ	3		
ミドリ	4		
シアン	5		
キイロ	6		
シロ	7		

❷**console**ステートメントで白黒モードが指定されているときは**color**ステートメントの第1パラメータは次

のような機能となる.

0 ノーマル　　　　4 リバース
1 シークレット　　5 リバース・シークレット
2 ブリンク　　　　6 リバース・ブリンク
3 シークレット　　7 リバース・シークレット

第2パラメータ，第3パラメータについてカラーモードのときと同機能である.

白黒モードでの画面の操作はcolorステートメントにより，いろいろな操作ができる.

次のプログラムを入力して実験してみてください．new RET をキー入力してからプログラムを入力してください.

プログラム1-6

```
10 console 0, 20, 0, 0
20 jnput x
30 color x, 0, 0
40 print "ガメンテスト"
50 goto 20
```

プログラムが入力できたら実行してみてください．入力は？ごとに0（ノーマル），2（ブリンク），4（リバース），6（リバース・ブリンク），3（シークレット），0（ノーマル）などを順々に入力して確かめてください.

コーヒータイム2

new RET キーを押してから次のプログラムを入力してください.

プログラム1-7

```
10 console 0, 20, 0, 1
20 c=rnd(1)*7+1
30 cc=int(c)
40 color cc, 233, 0
50 print cc;
60 goto 20
```

〔演習〕

このプログラムの1部を改造してあなたの名前をディスプレイに表示しましょう.

〔注意〕

①実行中のプログラムを停止するには STOP キーを使います.

②再実行する前に HOME CLR キーで画面クリアしましょう.

2 第2日 BASICプログラミング入門

〔目標〕

簡単な例題をもとに，パーソナル・コンピュータのプログラムは具体的にどのようにして作り，どのようにして実行し，結果を得るのかを知ります.

プログラムの書き方，マイコンへの入れ方，プログラムの確認の方法，修正の方法，実行のしかた，プログラムの停止法，および，プログラムのカセットへの保存方法，カセットから，プログラムをマイコンへ入れる方法など，プログラミングの全体について学習し，各人の学習進度に応じて2日目からでもプログラミングができるようにします.

1 BASICプログラムの書き方

第1日目ではプログラムの説明をしないで実験的に確かめています．ここでは，改めてプログラムの書き方について学びます.

プログラムの書き方としてはいろいろな方法がありますが，入門時からあまり特殊な形のものは覚えない方がよいでしょう．できるだけ，標準的，基本的な形を理解するようにしてください.

BASICプログラムの基本形は次のような形です.

```
行番号␣ステートメント RET
行番号␣ステートメント RET
      ⋮              ⋮
行番号␣ステートメント RET
行番号␣END RET
```

ここで，行番号は0～65529までの整数です．次の␣印はスペース記号です．ステートメントは，プログラムが実行されたときに実際に実行される命令文です．たとえば，PRINT文とかINPUT文などというものがあります.

RET は RETURN キーを表わします．このキーを押すことによって，初めて1行分のステートメントがメモリの中へ格納されます.

最初の頃はこの RET キーの意味が理解しにくいようですが，RET キーまでがひとつの命令文だと理解してください．このことは，ひとつの行番号に対するステートメントが，RET キーによって，初めて区切られ，他の行番号のステートメントと区別されます.

特に，ENDステートメントの最後の RET キーを忘れないようにしましょう.

では，次のプログラムを作って実行することにします.

〔例題1〕

A＝10，B＝2としてその和，差，積を求め，印刷するプログラムを作る.

【考え方】

①Aに10を入れる.	100␣ LET␣ A=10
②Bに2を入れる.	110␣ LET␣ B=2
③A，Bを書く.	120␣ PRINT␣ A, B
④X＝A＋Bを計算する.	130␣ LET␣ X=A+B
⑤Y＝A－Bを計算する.	140␣ LET␣ Y=A−B
⑥Z＝A×Bを計算する.	150␣ LET␣ Z=A*B
⑦答X，Y，Zを書く.	160␣ PRINT␣ X, Y, Z
⑧止まる.	170␣ END

上のような考え方を具体的にBASICのプログラムとして書いたものが，右側のプログラムです.

このプログラムの書き方として，行番号は必ずしも0から始まる必要はありません．ここでは100からにしていますが，これは任意です．そして，ここでは，10番飛びにし，次は110となっています.

あとで文の追加などがある場合にも便利なようにあらかじめ飛ばした行番号になっています．10番飛びなどが一般によく使われています.

行番号とステートメントの間はスペースを1個以上入れるのが標準形です．また，ステートメントと式A＝10，X＝A＋Bなどの間も一般にスペースを1個入れておいてください．

プログラムを書く前にはプログラムの左側に書いたような考え方というか“算法”といわれるものが確立されていなければなりません．

簡単なものでも1ステップずつ，きちんと書き表わすことが，結局は速くプログラムを完成させることになります．

考え方に少しでも不明なところがあったり，不完全なところがあると，プログラムは書けたとしても，実行できなかったり，答が間違っていたりします．そのときはまた最初に戻って考え直すということになるので，この段階はねばり強く問題をしっかり理解し，整理して，考え方をまとめることが大切です．

マイコンの命令は必要なときに調べ，使っていくうちに自然と使い方も身についてくるので，あわてず，ゆっくりと，着実に基礎事項を理解することに努めてください．

〔注意〕

ステートメントLETは省略できるので一般には省略形が使われています．

```
LET  A=10    ──→ A=10
LET  B=2     ──→ B=2
LET  X=A+B   ──→ X=A+B
```

などと書くことがよくあります．

〔演習〕 例題1のLET文を省略形で書いてください．

2 プログラムのマイコンへの格納法

問題の解法がまとまり，プログラムが書けたら，次はいよいよマイコンのメモリの中へプログラムを格納することになります．次にその手順を示します．

❶ HOME CLR キーを押して画面を消去してください．

❷ N E W のキーを押してNEWコマンドを実行し，以前のプログラムをすべて消去する（以前に作ってメモリに格納してあるプログラムを消去したくないときは，このNEWコマンドを使ってはいけません）．

❸ファンクション・キー f・2 を押し，最初の行番号が100なので 1 0 0 のキーを押して RET キーを押す．

この時点での画面は図2-1のようになっていることを確認してください．

❹ A，=，1，0 をキー入力する． RET キーを押す．

この時点での画面は図2-2のようになっていることを確かめてください．

❺以下同様に各行番号に相当するステートメントを入力していきます．

❻ENDステートメントを入力した後は，図2-3のようになり，ここで STOP キーを押してAUTO機能を解除します．

以上で，プログラムがメモリの中へ格納されたことになります．

それでは，本当に格納されているかどうか確かめるにはどうしたらよいのでしょうか．

〔注意〕

プログラムを格納中にキー入力などのミスをした場合には，

❶ INS DEL キーで抹消してください．

❷1行単位で間違えたときは，間違えたままにして RET キーを押し，次の行番号のステートメントを入力してください．

このときは，ENDステートメントを入力後，AUTO機能を解除した後，新たに，間違ったステートメントの行番号とステートメントを入力してください．

図2-1

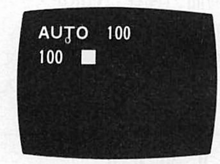


図2-2

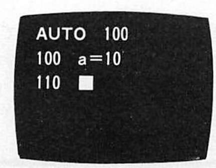
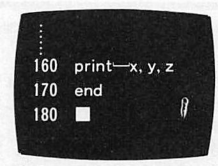


図2-3



3 プログラム・リストの確認法

マイコンのメモリへ格納されたプログラムのリストを確認したい場合がありますが，このときは，LISTコマンドを使って画面にプログラム・リストを表示させます．

2.2でプログラム入力後，AUTO機能を解除するため， STOP キーを押した後ではOkが表示されているので，ここで， L I S T または f・4 キーを押して RET キーを押してください．すると，先ほど小文字で入力したプログラムのリストが，大文字になって図2-4のように表示されます．

このように，マイコンのメモリの中にどのようなプログラムが入っているのかを確かめるには，このLISTコマンドを使えばよいのです．

1行だけのリストをみたいときには，《LIST行番号》で1

図2-4

```
100  A=10
110  B=2
120  PRINT  A, B
130  X=A+B
140  Y=A-B
150  Z=A*B
160  PRINT  X, Y, Z
170  END
```

行だけを表示することもできます．

いろいろなLISTコマンドの使い方を次にまとめておきます．

まとめ LISTコマンド

- メモリに格納されているプログラム全部を画面に表示する．……………………………**LIST**
- 行番号160のプログラムのみを表示したい．……………………………**LIST 160**
- 行番号130から最後の行番号までのプログラムを表示したい．……………………………**LIST 130—**
- 最も若い行番号から行番号150までのプログラムを表示したい．……………………………**LIST —150**
- 行番号120から行番号160までのプログラムを表示したい．……………………………**LIST120—160**

〔注意〕

❶**LIST**コマンド実行中では大きなプログラムは，画面に表示されてもスクロールにより消えてしまうので，途中で停止させたいときがあります．

このときは**ESC**キーを押します．再開するには，任意のキーを押します．

❷プリンタにプログラム・リストを印刷させたいときには**LLIST**コマンドを実行します．機能は**LIST**コマンドと同様です．

〔演習〕

まとめの**LIST**コマンド例について実験的に確かめてください．実験に先き立って**HOME CLR**キーで画面を消しておくとよいでしょう．結果は図2-5に示します．

図2-5

LIST	100 A=10 110 B=2 120 PRINT A, B 130 X=A+B 140 Y=A−B 150 Z=A*B 160 PRINT X, Y, Z 170 END
LIST 160	160 PRINT X, Y, Z
LIST 130—	130 X=A+B 140 Y=A−B 150 Z=A*B 160 PRINT X, Y, Z 170 END
LIST—150	100 A=10 110 B=2 120 PRINT A, B 130 X=A+B 140 Y=A−B 150 Z=A*B
LIST120—160	120 PRINT A, B 130 X=A+B 140 Y=A−B 150 Z=A*B 160 PRINT X, Y, Z

4 プログラムの修正

プログラムの作成途中や実験後などにプログラムの修正が必要になることがあります．むしろ，追加，訂正，変更などの必要があるのが普通のプログラムともいえます．

BASICのプログラムでは，この修正作業がきわめてやさしくできるところに特徴があります．ステートメントを行番号で管理しているので，行番号を中心にプログラムの修正作業ができます．

プログラムの修正方法としては2つの方法があります．ひとつは行単位で修正するもの，他のひとつは画面制御用キーを使用して，行の一部分を修正する方法です．

■行単位の修正方法

❶1行全体を消去したいとき．

消したい行番号のみを入力し**RET**キーを押します．たとえば，行番号140のプログラムを消したいときは，**140 RET**と入力してください．

❷新しい行を追加したいとき．

ある行とある行との間に新しい行を追加したいときにはその2つの行番号の間の数値を行番号としてプログラム文を追加します．

たとえば，行番号100と行番号110の間に追加したいときは，105の行番号を使うなどです．

プログラムの最後に新たに行を追加するときは，そのプログラムで使っている最大の行番号より大きな行番号を使います．

❸すでに格納してある行の内容を一部変更したいとき．

ⓐその行の全体を再度入力し直す方法．

たとえば，100 A=10とあるのを，

100 A=50としたいときは，

新たに，**100 A=50**とキー入力し直します．

ⓑ**スクリーン・エディタ**と呼ばれる画面制御用キーを使って部分的に修正する方法．

1文字消去，1文字挿入，シフト・キーなどの組み合わせによります（**第1日に学習**）．

〔注意〕

❶プログラムの修正を行なったら必ず，**LIST**コマンドでプログラム・リストを画面に表示させ，確認するクセをつけましょう．

❷プログラム修正作業で，特に**スクリーン・エディタ**による画面制御キーによる作業では，修正箇所の訂正，変更の後では必ず**RET**キーを押すことを忘れないでください．

この**RET**キーを押して，初めて，修正が行なわれたことになることを忘れないようにしましょう．

〔演習〕

図2-4のプログラム・リストを再度，画面に表示させ，プログラムに間違いがないか確かめましょう．特に，コンマや，演算記号などに注意してください．間違いがあれば直してください．

5 プログラムの実行

さて，いよいよプログラムを実行してみる段階になりました．次の手順で，とにかくプログラムを走らせてみましょう．

❶ HOME CLR キーを押して，画面を消す．
❷ f・4 ，RET キーを押して，プログラム・リストを表示する．
❸ f・5 のキーを押す．

図2-6



結果はどうなったでしょうか．画面には，図2-6のような結果が表示されるでしょう．

プログラムを実行させるには R U N とキーを押すことによって**RUN**コマンドを入力するです．f・5 はこのことを意味します．第1日目の学習を思い出してください．

ここで行なった，HOME CLR キーを押したり，f・4 キーによるプログラム・リストの表示はプログラムの実行には必要ありません．ただ，理解しやすくしたにすぎません．

RUNコマンドは行番号の最も小さいステートメントから実行を開始し，遂次，プログラムを実行して最後のEND文で終了します．

プログラムの途中から実行させたいときも，ときには必要となりますが，そのときは，実行させたい行番号を付けます．たとえば，行番号120から実行させたいときは**RUN 120**とキー入力します．

まとめ

行番号の最小のステートメントからの実行．**RUN** RET
プログラムの途中からの実行．**RUN 行番号** RET

〔演習〕

次に f・5 キーを何回も続けて押してみてください．その結果，画面にはどのようなものが表示されるか確かめましょう．行番号100から170のプログラムが何回も使われていることがわかります．

6 プログラムの停止

プログラム実行中に，実行を強制的に中止させたい場合があります．**プログラムで停止させる**には**STOP**文や**END**文を使いますが，次のようなプログラムでは永久にプログラムが走ってしまいます．途中で停止させるには ESC キーや STOP キーを使います．

図2-7は **プログラム2-1** の実行結果の一部です．

プログラム2-1

```
100  I=0
110  I=I+1
120  PRINT  I, 2*I
130  GOTO  110
```

図2-7

1	2
2	4
3	6
4	8
5	10
6	12
7	14
8	16
9	18
10	20
11	22
12	24
13	26
14	28
15	30
16	32
17	34

まとめ

❶実行中のプログラムを一時中止させたいとき，
ESC キーを押す．
再開させるには，任意のキーを押す．
❷実行中のプログラムを停止させるには，
STOP キーを押す．
再開は，**CONT**コマンド，**RUN**コマンド，**GOTO**コマンドによる．

〔演習〕

前記プログラムをメモリに格納し，実行させ，ESC キーと STOP キーの機能を確認してください．

〔注意〕

このプログラムを格納する前に**NEW**コマンドを使って前のプログラムを消去してから格納してください．



プログラム2-2

```
100  FOR  I=33  TO  255
110  PRINT  CHR$(I);
120  NEXT  I
130  PRINT :PRINT :PRINT
140  GOTO  100
```

上のプログラムをメモリに格納し，実行してみてください．画面には，図2-8のようなキャラクタのコードが表示されます．PC-8001では，これだけの文字，記号，パターンが，使えることがわかります．

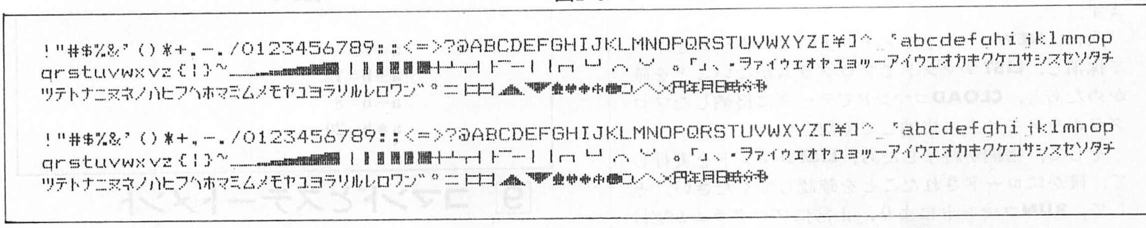
プログラムを停止させたいときは，ESC キー，または，STOP キーを使ってください．

7 プログラムの保存法

いままでいろいろなプログラムをメモリに格納しては，実行し，また新しいプログラムをメモリに格納するときには，**NEW**コマンドを使って，せっかく作って格納したプログラムを抹消してしまっていました．

一度作ったプログラムを再び使えるようにどこかへしまっておくと，とても便利になります（マイコンのメモリに

図2-8



格納してもよいのですが，プログラムが格納されるメモリは電源を切るとプログラムがすべて壊れてしまい，再び電源を入れて**RUN**コマンドを実行してもプログラムを実行するこはできません).

マイコンではフロッピーディスクなどの高価な外部記憶装置もありますが，カセットテープレコーダを使ってカセットテープへも格納できるので，この本で紹介するプログラムはすべて，カセットテープへしまったり，カセットテープから出してきてマイコンのメモリへプログラムを再入力したりすることにします.

●プログラムをカセットテープへ格納する方法(セーブ)

カセットを録音状態にしておいて，**csave"プログラム名"** **RET** とキーを押します．すると，カセットは録音を開始し，行番号の最も小さなステートメントから順番に最後の行番号までのプログラムをすべて格納して，停止します.

【例】

csave "PR-1" RET

メモリ上にあるプログラムをカセットテープにファイル名 **"PR-1"** によって格納します.

ここで，プログラム名は，6文字以内で作ってください．名前は任意です.

〔注意〕

❶カセットを録音状態にしたときに，駆動する場合は**motor**とキー入力して停止してから**csave**コマンドを実行してください.

❷録音開始時には，カセットの**カウンタ**をゼロにし，録音完了後の**カウンタ**の値を記録しておくとメモリに格納するときに目安がつき便利です.

●録音状態を確認する(ベリファイ).

カセットへ録音したときはプログラムが確かに格納されたか確認する習慣をつけておくと，録音ミスを防ぐことができます.

まず，カセットテープを録音開始時のカウンタの値まで戻します．カセットテープは再生状態にしておきます．**cload? "プログラム名"** **RET** とします.

【例】

cload? ␣"PR-1" RET

メモリ上にあるプログラムとカセットテープに格納されているファイル名 **"PR-1"**のプログラムとが比較され，一致しているかどうか調べます.

万が一，うまく録音されていなかったときは**Bad**と表示されます．うまく録音されているときは**Ok**と表示され，コマンド入力モードになります.

●テープの内容とメモリ上のプログラムの内容が一致した場合は，

```
cload? "PR-1" RET
Found PR-1
Ok
■
```

と表示されます.

●テープの内容とメモリ上のプログラムの内容が一致しない場合は，

```
cload? "PR-1" RET
Found  PR-1
Bad
Ok
■
```

と表示されます.

この場合はうまくテープにプログラムが録音されなかったので，もう一度**CSAVE**コマンドを実行して録音し直してください.

Found "プログラム名" が表示されない場合も同様に考えられるので，**CSAVE**コマンドをもう一度やり直してください.

●プログラムをカセットテープからメモリ上へ格納する(ロード).

ベリファイが完全に行なわれたテープには，プログラムが正しく格納されていることになります．カセットテープに格納されているプログラムをメモリ上に格納するには**CLOAD**コマンドを用います.

【例】

cload "PR-1" RET

格納中は画面の右上に＊印が点滅します．当然，カセットは再生状態でこのコマンドが実行されなければなりません．カセットからのプログラム格納が完了すると**Ok**が表示され，コマンド入力モードになります.

```
cload "PR-1" RET      *
Found  PR-1
Ok
■
```

まとめ

プログラムのセーブ，ベリファイ，ロード

●テープに格納する	**csave "プログラム名"** (6文字以内)
●テープ格納状態の確認	**cload? "プログラム名"**
●テープからメモリ上への格納	**cload "プログラム名"**

〔演習〕

適当なプログラムをカセットにセーブし，ベリファ

イを実行して，録音状態が間違いのないことを確かめます．

次に，**NEW**コマンドでメモリ上のプログラムをすべて抹消し，**LIST**コマンドでプログラムがないことを確かめたあと，**CLOAD**コマンドでテープに格納したプログラムをメモリ上に格納してください．

そして，格納が終了したら，**LIST**コマンドを実行して，確かにロードされたことを確認してください．そして，**RUN**コマンドにより，正常にプログラムが実行されることを確めてください．

〔注意〕

カセットテープにプログラムを格納したときは，必ずプログラム名とカウンタの値（初期値と最終値）をそのカセットテープに記入しておいてください．

8 プログラミングの考え方

プログラムの書き方は，必ずしも1つとは限りません．10人がそれぞれ書けば，その中にはいくつか違ったものがあるのが普通です．

たとえば，次のプログラムのように書いてもA, B, X, Y, Zをプリントします．

プログラム2-3

```
100  A=10:B=2
110  X=A+B:Y=A-B:Z=A*B
120  PRINT  A, B, X, Y, Z
130  END
```

●マルチステートメント

先のプログラムでは，ひとつの行番号に2つ以上のステートメントが書かれています．ステートメントの区切り記号：が行に含まれている場合は，**マルチステートメント**になります．行番号100と行番号110が**マルチステートメント行**です．

大きなプログラムになるようなら，マルチステートメントのプログラムにした方がメモリ使用効率がよくなるのですが，小さなプログラムでメモリ容量に余裕がある場合は，できるだけマルチステートメントをさけ，わかりやすい書き方のプログラムにしておいた方がよいでしょう．

●計算結果をみやすくするには

プログラム2-3の出力結果だけでは，その意味することがわかりません．そこで，**プログラム2-4**に示すような書き方もあります．PRINT文の中に“ ”で囲んで注釈をいれると出力結果が見やすくなり，図2-9のように意味もわかりやすくなります．

引用符で囲まれた文は**プロンプト文**といわれ，そこに記入された文のまま，ディスプレイ上に表示されます．

プログラム2-4

```
100  A=10:B=2
110  X=A+B:Y=A-B:Z=A*B
120  PRINT  "a=";A, "b=";B
140  PRINT  "a+b=";X
150  PRINT  "a-b=";Y
160  PRINT  "a*b=";Z
170  END
```

図2-9

```
a=10
a+b=12
a-b=8
a*b=20
```

9 コマンドとステートメント

いままで，詳細な説明なしに**コマンド**とか**ステートメント**などという言葉を使用してきましたが，ここで，少しだけ説明しておきます．

コマンドは，ユーザーがBASICプログラムを作るときに使用する，マイコンのモニタ・プログラムに対する指令であって，普通はマイコンを使うのに便利なように準備されています．

PC-8001での代表的な**コマンド**には，**LIST, RUN, NEW, AUTO, CSAVE, CLOAD?, CLOAD**などがあります．ディスプレイ上で**OK**が表示された後はすべて，これらの**コマンド**を受け付けることができる状態になります．

ステートメントは，ユーザーのプログラムの中で使われる文で，普通は行番号の次に書かれます．

代表例として，PRINT, END, GOTO, LETなどで，PC-8001ではその他，たくさん用意されています．

10 動作モード

PC-8001では，3つのモードがあり，いずれかの動作モードにあります．

●間接モード（プログラム・モード）

行番号␣BASICステートメント RET の形でプログラムをメモリに入力して，プログラムを実行できるモードです．入力したプログラムは，何回でも**RUN**コマンドにより実行させることができます．

●直接モード

行番号を付けずにコマンドや，ステートメントが使えるモードで，入力されたら RET キーですぐ実行されます．**RUN**コマンドで実行させるのではないことに注意してください．命令は，実行後失なわれるので，数回実行させるには，数回命令を入力しなければなりません．

【例1】

?3+5 RET とキー入力すると8と表示されます．

【例2】

A=3:B=5 RET とキー入力し，続いて，
?A+B RET とすると8と表示します．
?A*B RET とすると15と表示します．

●ターミナル・モード

PC-8001を，コンピュータの端末として使用するときの動作モードで**TERM**コマンドにより，このモードになります．

〔演習〕

ダイレクト・モードで，次のステートメントを実行してください．

❶a=10：b=2
❷print　a，b
❸print　"a，b"
❹print　a+b
❺print　"a+b"
❻print　"a+b=";a+b

〔演習〕
前のプログラムを改良して，A÷Bの計算値も印刷できるようにしてください．割算の記号は／（スラッシュ）を使います．

コーヒータイム 4

次のプログラムを入力して，実行してみてください．このプログラムは三角関数表の作成プログラムです．

印刷形式を指定せずにプログラムしたものです．結果を図2-10に示しました．

SIN(X)，COS(X)，TAN(X) はそれぞれ**X**の単位をラジアンとした場合の三角関数の値です．演算は単精度で，7桁までの精度で格納され，6桁までの桁数で印刷されます．

プログラム2-5

```
10 PRINT"X    sin(x)   cos(x)   tan(x) "
20 PRINT
30 FOR X=1 TO 10
40 PRINT X;SIN(X);COS(X);TAN(X)
50 NEXT X
60 END
```

図2-10

```
X    sin(x)     cos(x)     tan(x)

1    .841471    .540302    1.55741
2    .909298   -.416147   -2.18504
3    .14112    -.989993   -.142547
4   -.756802   -.653644    1.15782
5   -.958924    .283662   -3.38052
6   -.279416    .96017    -.291006
7    .656987    .753903    .871448
8    .989358   -.1455     -6.79973
9    .412118   -.91113    -.452316
10  -.544021   -.839072    .64836
```

3 第3日 データの与え方と表示

〔目標〕
プログラムで使われるデータ（数値や文字）をどのようにしてマイコンに与えるか，いろいろなデータの与え方を理解するとともに，データの出力についても同様にいろいろな出力形式があるので，これらについても理解します．

出力についてはCRTディスプレイ上における表示について勉強し，プリンタ上への印刷については少しふれる程度にします(プリンタの性能によるところが多いため)．

1 画面とプリンタへの出力

CRTの画面上へデータを表示させたいときは，一般にPRINT文を使います．

たとえば，

PRINT "ABC"

とすれば，CRT画面に

ABC が表示されます．

次に，プリンタに印刷させたいときは，次のようにPRINT文をLPRINT文にします．

LPRINT "ABC"

とすれば，プリンタに

ABC

が印刷されます．

以上のようにCRT画面上への表示と，プリンタの用紙上への印刷出力は**PRINT**と**LPRINT**の違いで区別されます．ます．

〔演習〕 プリンタへの出力実験

プリンタを持っている人は次のプログラムを実行してみてください．このプログラムは前の**プログラム2-2**の**PRINT**文××を改造したものです．

プログラム3-1

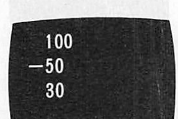
```
100 FOR I=33 TO 255
110 LPRINT CHR$(I);
120 NEXT I
130 END
```

2 数値の出力

いま，3つの数値A＝100，B＝50，C＝30が与えられ，これらの数値を画面に出力することを考えてみます．

● 1行に1つの数値を出力する場合

プログラム3-2

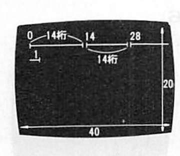



```
10 A=100:B=-50:C=30
20 PRINT A
30 PRINT B
40 PRINT C
50 END
```

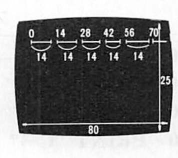
PRINT Aのように，**PRINT**文がコンマやセミコロンで終っていない場合は改行されます．

● 標準化された位置に，続けて出力する場合

PC-8001では，**WIDTH**文の桁数指定により，表示開始位置が14桁ごとの位置に標準化されていて，**PRINT**文中のコンマの直後に書かれている項目の表示開始位置は自動的に14桁ごとの位置になります．



WIDTH 40, 20 の場合

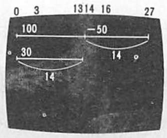


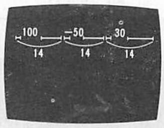
WIDTH 80, 25の場合

プログラム3-3

```
100  A=100：B=−50：C=30
110  PRINT A,B,C
120  END
```

左のようなプログラムでは WIDTH 40, 20のとき2行にわたってしまうがWIDTH 80, 25のときは1行になります。





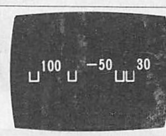
この場合は自動的に実行されます。

●1行に続けて出力したい場合，

間隔をとらずに直前の出力結果のすぐ後に続けて出力したい場合には，PRINT文中にセミコロンを使います．

プログラム3-4

```
100  WIDTH  40, 20
110  A=100：B=−50：C=30
120  PRINT  A；B；C
130  END
```



数値出力は直後に空白（スペース）が1つ挿入されます．また，負の数値はマイナスの符号（−）が加えられます．正の数値のときは符号が出力されず，空白が1つ出力されます．

行番号120の文を次のようにしても同じように出力されます．

```
120  PRINT  A；B；
125  PRINT  C
```

120行の終りのセミコロンにより，125行のプリント文が同じ行に出力されます．

●行と行の間をあけたい場合，

1行出力したあと，次の行との間に1行分の空白行を挿入したいときは，キーワードのPRINTだけを使います．

プログラム3-5

```
10  A=100：B=−50：C=30
20  PRINT  A
25  PRINT
30  PRINT  B
40  PRINT  C
50  END
```



上のプログラムでは25行のプリント文により空白行が1行できます．

2行分空けたいときはPRINT：PRINTとします．

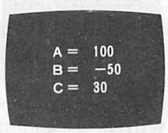
●メッセージや説明文を出力したい場合，

前述のプログラムのようにAやB，Cの値をただ単に，100，−50，30と出力するよりも，A=100とか，B=−50，C=30と出力したい場合があります．

このようなときは，キーワードのPRINTにつづき，引用符" "の中に出力したいメッセージまたは，説明文などを入れておきます．

プログラム3-6

```
10  A=100：B=−50：C=30
20  PRINT "A="；A
30  PRINT "B="；B
40  PRINT "C="；C
50  END
```



左のプログラムを実行すると，右図のような出力が得られます．引用符で囲まれた文を，**プロンプト文**といいます．

3 数値の入力

数値を与えるのに代入文でA=100，B=−50，C=30のように記述してきました．ここでは，他にいくつかの方法があるので，それらについて考えてみましょう．

●キーボード入力

代入文で数値を与えることのほかに，キーボードから直接，そのつど，数値を与える方法があります．そのためにはキーワードINPUTが使われます．

```
10  INPUT  A        10  INPUT  A, B, C
20  INPUT  B        (この場合は1度にA，B，Cの値を
30  INPUT  C         キー入力した後リターン・キーを
                     押す．)
```

上のものは左も右も結果は同じでA，B，Cにそれぞれキーボードから入力された数値が代入されます．

INPUT文が実行されると"？"が出力され，キー入力を要求されるので，対応するキー入力操作をすることになります．この例では，最初の"？"で100を，次の"？"で−50を，最後の"？"で30をキー入力すると，A=100，B=−50，C=30の代入文と同じことになります．

〔演習〕

```
10  INPUT  A, B, C
20  PRINT
30  PRINT  A；B；C
40  END
```

上のプログラムを実行させてみてください．

●データを間違いなく入力するには，

INPUT文にどんな入力データかを入力要求とともに表示させるとよいでしょう．そのためにキーワードINPUTに続いて引用符" "で囲んだ説明文を入れるとよいでしょう．

```
10  INPUT  "A="；A
20  INPUT  "B="；B
30  INPUT  "C="；C
```

〔演習〕

```
10  INPUT  "a=";A            } 入力
20  INPUT  "b=";B
30  W=A+B       ……足し算   }
40  S=A−B       ……引き算   } 演
50  M=A*B       ……掛け算   } 算
60  D=A／B      ……割り算   } 処
70  V=A^B       ……べき乗   } 理
80  PRINT W;S;M;D;V          }
90  PRINT  "a+b=";W          }
100 PRINT  "a−b=";S          } 出力
110 PRINT  "a*b=";M          }
120 PRINT  "a／b=";D         }
130 PRINT  "a^b=";V          }
140 PRINT                    空白行
145 GOTO   10
150 END
```

左のプログラムを格納して実行してみてください．いろいろな入力データをA，Bに与えて出力結果を確かめてください．80行と90-130行の違いをよく理解してください．140行は空白行です．

〔演習〕次の各ステートメントの違いを実験的に確かめましょう．

```
200  A=1：B=2：C=3
210  PRINT  A；B；C
220  PRINT  A，B，C
230  PRINT  "A，B，C"
240  PRINT  A+B+C
250  PRINT  1+2+3
260  END
```

●**固定のデータを扱う場合,**

プログラムを実行させるたびにキーボードからデータを入力する方法はそのつどデータが変わる場合に使います．変わらないデータはあらかじめプログラムの中へ格納しておく方法があります．この方法は**DATA**文を使うもので，多量のデータを与えるときにもよく使われます．

〔例〕

```
100  READ  A, B, C, D
110  PRINT  A, B, C, D
120  DATA  100, 200, 300, 400
130  END
```

このプログラムでは変数*A*には100, *B*には200, *C*には300, *D*には400が**READ**文によって代入されます．

DATA文はコンマで区切ってデータを並べておきます．

すなわち，**READ**文は**DATA**文で与えられたデータを順番に変数の並びに代入していくわけです．

〔例〕

```
100  READ   A, B
110  PRINT   A, B
120  DATA   100, 200, 300, 400
130  END
```

このプログラムでは*A*=100, *B*=200が代入され，**DATA**文で与えられた300, 400はどこにも代入されません．

〔例題〕

```
 90  FOR  I=1  TO  3
100  READ   A, B
110  PRINT  A, B
115  NEXT   I
118  END
120  DATA   100, 200, 300, 400
130  DATA   500, 600, 700, 800
```

このプログラムを実行して，**READ**文と**DATA**文の関連を理解してください．まず第1回目の**READ**文で，100, 200が*A*, *B*に，第2回目の**READ**文で300と400が*A*, *B*に，第3回目の**READ**文の実行で500と600が*A*, *B* にそれぞれ代入されています．

このように，**READ**文の実行回数により変数に代入されるデータ数が多い場合はエラーはありませんが，もし，データ数が少ない場合は「Out of Data」のエラーメッセージが画面に表示されます．

次のプログラムを実行して確かめてください．

〔演習〕

```
 90  FOR   I=1   TO   3
100  READ   A, B
110  PRINT   A, B
115  NEXT   I
118  END
120  DATA   100, 200, 300, 400
```



プログラム3-7

```
 10 PRINT DATE$ ; " " ; TIME$
 20 PRINT "アサヒ ブンカ センター"
 30 PRINT
 40 G=0
 50 CONSOLE 0, 20, 0, 1
 60 COLOR 6, 32, 0
 70 LINE INPUT "ショウヒン メイ=" ; A$
 80 COLOR 7, 32, 0
 90 INPUT "タンカ=" ; U
100 INPUT "スウリョウ=" ; N
110 C=U*N
120 G=G+C
130 PRINT "キンガク=" ; C
140 COLOR 5, 0, 0
150 PRINT "ゴウケイ=" ; G ; "円"
160 COLOR 7, 0, 0
170 PRINT
180 GOTO 50
```

4 文字の入出力

2, 3では数値の出力，入力について勉強しました．実は文字についても数値と同様，マイコンで入力，出力できます．

●**文字変数の作り方**（INPUT X$）

数値と同じように文字にも変数型があります．次のプログラムを実行しながら理解しましょう．

〔演習〕

```
200  INPUT   A$      ……キーから文字列を入力．
210  PRINT   A$
220  GOTO   200
```

文字変数は数値変数で使える変数名の後に **$**（ドルマーク）を付ければ文字が変数として取り扱かえます．**A$** は文字変数です．代入文で与えるには，

A$= "文字列"

とします．

〔演習〕　**プログラム3-8　文字出力**

```
250  A$="ナゴ   ヤシ"
260  B$="ミズ   ホク"
270  PRINT   A$, B$
280  PRINT   A$ ; B$
290  PRINT   A$B$
300  PRINT   A$B$"I/O"
310  END
```

250, 260でA$の文字変数にナゴヤシが，B$の文字変数にミズホクが代入されます．文字変数をPRINTする場合には，セミコロン（;）がなくても290のように続けて出力してくれます．

> **プログラム3-8で，RUN CR のあと，**入力文字列の中に，コンマ（,）や引用符（"）を混ぜて，入力してみてください．**INPUT**文では，これらの文字を入力することはできません．
>
> プリント結果では数値出力のようにデータの前後に空白ができないことを確認してください．
>
> 入力できる文字列の長さは255文字以内です．

●**文字列の入力**（LINE INPUT）

とにかく，マイコンで取り扱かえるすべての文字（コンマや引用符も含めて）を入力するには**LINE INPUT**文を使います．

プログラム 3-9

```
350 LINE INPUT L$
360 PRINT "*"L$"*"
370 GOTO 350
```

〔演習〕

次のプログラムを実行させて，READ，DATA文による数値入力，文字入力の方法を理解しましょう．

```
100 READ A, B
110 READ C$, D$
120 PRINT "A="; A, "B="; B
130 PRINT "C$="; C$, "D$="; D$
140 DATA 100 , 200 , ABC , DEF
150 END
```

プログラム3-9を**RUN350**で走らせ，空白なども含めて文字列を入力してみてください．＊印で囲まれたものが入力され，出力された文字列です．

LINE INPUT文では，カンマ，引用符などすべての文字を入力できますが，文字列のうしろの空白は無視されます．

また，**INPUT**文では，**INPUT A$，B$，C$**などと，複数の文字変数の並びが許されましたが，**LINE INPUT**文では，文字変数の並びは１つです．

それから，**INPUT**文との違いは〝?〟マークが画面に表示されないことです．

●キー入力した文字を画面に表示させない方法(INKEY$ INPUT$)

INPUT文や**LINE INPUT**文では，入力した文字を画面に表示しますが，表示したくない場合があります．

そのようなときには**INKEY$**（１文字入力用），**INPUT$(X)**（X個の文字入力用）という関数を使います．

この関数は **RETURN** キーを押さなくても入力できるので，入力のつど **RET** キーを押すわずらわしさを解消したい場合にもよく使われます．

〔例〕　　プログラム 3-10

```
400 PRINT"INKEY$ TEST"
410 K$=INKEY$
420 1F K$=" " THEN 410
430 PRINT K$
440 PRINT
450 GOTO 400
```

INKEY$という関数は，キーが押されればその文字を文字変数に与え，押されない場合は，その値は空（ヌル・ストリング）となります．

420の**IF**文は，判断文で，もし**K$**の値が空ならば，行番号**410**へ飛べ，という意味です．もし**K$**の値が空でなければ，次の文，**430**の**PRINT**文を実行します．

〔例〕　　プログラム 3-11

```
500 PRINT" INPUT$ TEST"
510 P$=INPUT$(6)
520 PRINT P$
530 PRINT
540 GOTO 500
```

INPUT$**という関数は**INPUT$(N)という一般形で*N*個の文字を入力することを意味します．*N*個のキーを押し終ると，*N*文字が**P$**に代入されます．**プログラム3-11**は６文字入力の例です．

●ファンクション・キーによる入力（KEY）

いくつかの文字をキー入力する場合，文字の個数だけキーを押す方法だと，何度も何度も同じ綴りの文字列を入力するのが面倒になります．

このようなとき15文字以内なら，ファンクション・キーを利用する方法があります．これは**KEY**定義文によって入力したいデータを定義しておけばよいわけです．

キー定義文は，

```
KEY 1, "アサヒ ブンカセンター" +CHR$(13)
```

という形式です．**CHR$(13)**は入力時に **RETURN** キーを押さなくても，ファンクション・キー **f･1** だけ押せば，〝アサヒブンカセンター〟という文字列が入力でき，**RET** キーを押す操作を省略できます．

〔例題〕

```
550 KEY 1, "シイレ プログラム" +CHR$(13)
560 KEY 2, " ウリアゲ プログラム"+CHR$(13)
570 KEY 3, "ザイコ カクニン"+CHR$(13)
580 INPUT "シイレ(f･1),ウリアゲ(f･2), ザイコ(f･3)"; X$
590 PRINT X$ : PRINT
600 GOTO 580
```

〔注意〕

上の**プログラム**を実行する前に，その時点での**KEY**定義がどんな内容か，確認しておいてください．

確認には，ファンクション・キーの内容をリストできる**KEY LIST**文をダイレクト・モードで実行します．

まとめ　データの入力

マイコンにデータを与える方法には次の３つの方法がある．

❶キー入力操作による方法

①INPUT文による方法

(例)	INPUT A	数値入力
	INPUT A$	255文字以内

②LINE INPUT文による方法

(例)	LINE INPUT X$	255文字以内

③INKEY$による方法

(例)	X$=INKEY$	1文字入力

④INPUT$による方法

(例)	X$=INPUT$(N)	*N*文字入力

⑤ファシクション・キーによる方法

(例)	KEY1, "YES"+CHR$(13)	15文字以内入力

❷プログラムに書いておく方法

①代入文による方法

(例)　A=10
　　　A$="ABC"

②READ文とDATA文による方法

(例)　10 READ A, A$
　　　20 DATA 10, ABC

❸その他，カセットテープなどからの入力方法がありますが，これらについては後で説明します．

5 表示位置の指定

●カーソルの移動（LOCATE文）

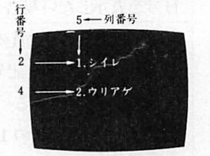
画面の任意の位置に文字を表示したいときは，**LOCATE**文を使います．

LOCATE文を実行するとカーソルは指定された位置に移

動するので，そのあと，PRINT文を実行すれば指定位置から表示するわけです．

〔演習〕

```
10 LOCATE 5, 2
20 PRINT "1. シイレ"
30 LOCATE 5, 4
40 PRINT "2. ウリアゲ"
50 END
```



LOCATE文には，カーソル・スイッチがあり，上述のプログラムのように省略するとカーソルは画面に表示されませんが，LOCATE5, 2, 1のようにすると，カーソルが指定した位置に現われます．

●カーソルの取り扱い

(1)カーソルの現在位置の列番号を知るには，POS(J)を使います．ここでJはダミーです．

(2)カーソルの現在位置の行番号を知るには，CSRLINの2つの関数を使います．

現在位置から上下左右へのカーソルの移動プログラムは次のようになります．

上方向はLOCATE POS(J), CSRLIN−1
下方向はLOCATE POS(J), CSRLIN+1
左方向はLOCATE POS(J)−1, CSRLIN
右方向はLOCATE POS(J)+1, CSRLIN

●スペースやタブを使った出力

PRINT文中での空白や位置制御のためには，SPCやTAB関数を使います．

プログラム3-12

```
100 PRINT "コード"; TAB(10); "ヒンメイ"
110 PRINT
120 READ A$, B$
130 PRINT A$; TAB(10); B$
140 DATA 111 , メロン
```

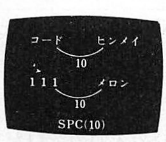


TABは指定した位置まで空白を出力します．

〔演習〕

プログラム3-12の行番号100と130のTAB(10)をSPC(10)に置き換えてRUN100を入力して実行してみましょう．

SPC関数は指定した数だけの空白をプリント出力します．



〔演習〕

TABとSPC関数の違いを理解してください．

TABは表示されている文字と同じ行へ出力しても文字を消去しないが，SPCは表示されている行へ再びPRINTすると，表示されていた文字は消去されます．

6 出力形式の編集 (PRINT USING文)

桁数を揃えたり，右詰めで出力したりして，出力結果を見やすく編集する方法がいくつかあります．それには，PRINT USING文を使います．

一般形は，

PRINT USING "<書式仕様>"; <出力並び>

です．

●桁数を揃えて右詰めで出力したい場合（＃）

引用符（〝）の中に〝＃〟記号を桁数だけ並べます．

〔例題〕

```
10 INPUT A, B, C
20 PRINT
30 PRINT USING "＃＃＃＃＃"; A
40 PRINT USING "＃＃＃＃＃"; B
50 PRINT USING "＃＃＃＃＃"; C
60 GOTO 10
```

このプロプログラムを実行させ，INPUT文で5桁以内の数値を入力して出力結果を確かめてください．

たとえば，Aに100，Bに−123，Cに3255.82を入力したとすれば，結果は，右図ように表示されます．すなわち，



❶〝＃〟記号で指定された桁数内で，数値は右詰めで表示されます．

❷頭部の余った桁には，空白が表示されます．

❸表示される数字の右端桁は，その下の桁を4捨5入して丸めたものが表示されます．

❹出力表示データが，〝＃〟の数より多い場合には，数値の直前に〝%〟が表示され，表示が異常であることを示します．

〔演習〕

次のプログラムを格納して実行させてみましょう．

80行のPRINT文と90～130行のPRINT USINGとの違いを理解しましょう．

```
10 INPUT "a="; A
20 INPUT "b="; B
30 W=A+B
40 S=A−B
50 M=A*B
60 D=A/B
70 U=A∧B
80 PRINT W; S; M; D; U
90 PRINT USING "a+b=＃＃＃＃＃"; W
100 PRINT USING "a−b=＃＃＃＃＃"; S
110 PRINT USING "a*b=＃＃＃＃＃"; M
120 PRINT USING "a/b=＃＃＃＃＃"; D
130 PRINT USING "a∧b=＃＃＃＃＃"; V
140 PRINT USING
145 GOTO 10
150 END
```

PRINT USING文では，引用符で囲まれた中に文字列があるとそのまま表示もできるので，次のようなこともできます．

```
200 Y=1981 : M=5 : D=31
210 PRINT USING"＃＃＃＃年␣＃＃月␣＃＃日"; Y, M, D
270 END
```

〔注意〕

```
PRINT USING "#####"; A, B, C
```

のように，書式仕様が不足した場合は同じ書式仕様が繰り返し使われ，数値変数B，Cも5桁以内で出力されます．

●小数点（．）を付けて出力したい場合，

書式仕様の中で，任意の位置に小数点1個を挿入できます．小数点以下の余った桁にはゼロが表示されます．

〔例〕

```
250  X=23.846 : Y=5.3 : Z=0.045
260  PRINT USING"␣##.###"; X, Y, Z
270  END
```

このプログラムの出力結果は，

```
␣23.846 ␣␣5.300 ␣␣0.045
```

となります．

●円記号（￥）を付けたいとき，

￥100とか￥360などのように数字の直前に〝￥〟を付けるには，2個の円記号（￥￥）を書式仕様の中で〝#〟記号の前に付けます．

〔例〕

```
300  INPUT  A
310  PRINT  USING  "¥¥###"; A
320  PRINT
330  GOTO  300
```

たとえば，360を入力すると，␣␣￥360と表示されます．

●3桁ごとにコンマ（，）で区切りたい場合，

千の位，百万の位などにコンマをつけて読みやすくするには，〝#〟記号の直後にコンマを書きます．小数点がある場合には小数点の直前にコンマを書きます．

〔演習〕

```
350  INPUT  B
360  PRINT  USING  "####,"; B
370  PRINT
380  GOTO  350
```

たとえば，4536を入力すると，4,536と表示されます．

〔演習〕

360行の引用符の中を####，.##として2456.7を入力してみましょう．

〔演習〕

3456と入力したら￥3,456と表示するプログラムを考えましょう．

```
400  INPUT  A
410  PRINT  USING  "¥¥####,"; A
420  PRINT
430  GOTO  400
```

〔注意〕

大きな桁数の数値を入力するには，次のような型宣言文字を使います．

変数の型	桁　　数	型宣言文字	例
倍精度型	8桁以上	#	A#
単精度型	7桁	!	A!
整数型	−32768～+32767	%	A%

書式記号としては，以上説明したもののほかに以下のような記号があり，それぞれの処理ができます．

機　　能	書式記号	例
符号を表示したい場合	+	+##，##+
〃　　　（負のときだけ）	−	##−
頭部の余った桁を*記号で埋める	**	**##
指数型の表示にする	∧∧∧∧	###∧∧∧∧
文字データの最初の1文字だけを表示する．	!	"!"; A$
文字データの表示桁数を指定．	&　&	"&&"; A$
注．！と&の記号は文字データ用の書式記号です．		

7 カセットテープによるデータの入出力

いままでは，画面へのデータ出力，キーボードからのデータ入力を説明してきましたが，データをカセットテープへ記録（出力）したり，カセットテープからデータを読み込む（入力）ことについて説明します．

●カセットテープへのデータの書き込み（出力）

ステートメントは

```
PRINT  #−1, <出力並び>
       ~~~~
         └カセットテープを意味する．
```

という形式です．

〔例〕　　　　プログラム3-13

```
100  A=1234 : A$="ABC"
110  INPUT "SAVE OK"; R$
120  PRINT #−1, A, A$
130  END
```

このプログラムの120行よりAとA$がテープに読み込まれ，1234とABCが記録されます．

〔注意〕

文字列の入力で，先頭に意味のある空白があるときと文字列の中にコンマが含まれているときは，CHR$(34)(これは引用符を意味する）で囲んで書き込んでください．

〔例〕

```
140  A$="␣␣ABC" : B$="D, E" : C$="FGH"
150  PRINT #−1, A$, C$
160  PRINT #−1, A$ ; C$
170  PRINT #−1, A$
180  PRINT #−1, CHR$(34) ; A$ ; CHR$(34)
190  PRINT #−1, B$
200  PRINT #−1, CHR$(34) ; B$ ; CHR$(34)
210  END
```

160行のPRINT#文では，A$とC$がセミコロンで区切られているので，ABCFGHのように連続した文字列としてテープに書かれ，また，このテープを読むときには，ABCFGHという1個のデータとして読まれます．

170行では，A$の先頭の空白が書き込まれないで，ABCだけが書き込まれます．180行のように両側にCHR$(34)があるときには，A$の先頭の空白も書き込まれます．

190行のPRINT#文では，DとEが分離されて2個のデータとして書き込まれます．しかし，200行では，3文字の列D，Eとして分離されず1個のデータとして書き込まれます．

●カセットテープからのデータの読み出し（入力）

ステートメントは

```
INPUT #−1, 〈入力並び〉
```

という形式です．ここで，INPUT＃文の入力並びはPRINT＃文の出力並びと対応します．

すなわち，個数，順序，データの型（文字型か数値型）が，一致していることです．

```
PRINT#−1,   A, A$, B, B$
              ↓   ↓   ↓   ↓
INPUT#−1,   C, C$, D, D$
```

変数名は必ずしも一致していなくてもよい．

〔演習〕

```
250  INPUT "LOAD OK" ; L$
260  INPUT #−1, B, B$
270  PRINT B, B$
280  END
```

〔例〕

あらかじめテープにデータを書き込んでおいてから，テプを巻き戻し，**プログラム3-13**を実行してみてください．画面に1234とABCが表示されることを確認してください．

コーヒータイム 6

PRINT USING文の復習です．複利計算での元利合計を画面に表示します．

10,000円を利率8.2％で5年間預けたらいくらになるでしょう．

プログラム　3-14

```
10  REM *** ガンリ ゴウケイ ***
20  INPUT "ガンキン ハ イクラデスカ"; X#
30  INPUT "キンリ ハ ナン% デスカ"; R
40  INPUT "ナンネン マデ ミマスカ"; Y
50  PRINT : PRINT
60  PRINT "ガンキン=";
70  PRINT USING"#########,"; X#;
80  PRINT "円"
90  PRINT "キンリ ="; R ; "%"
100 PRINT
110 PRINT "年"; TAB(7); "ガンリゴウケイ"
120 FOR A=1 TO Y
130 C#=INT (10*X#*(1+R/100)^A)/10
140 PRINT A ; TAB(6);
150 PRINT USING "##########,"; C#;
160 PRINT "円"
170 NEXT A
180 PRINT : PRINT : GOTO 20
```

第4日　多項目のデータの取り扱い方(配列)

〔目標〕

顧客管理とか，会計処理，実験データ処理などでは，多量のデータが取り扱われるとともに，その種類も多項目です．たとえば，顧客1人について，住所，電話番号，職業，年令，性別などの項目があります．また，毎月の会計処理では，月々の交通費，食費，消耗品費，会費，図書費などの項目があります．

これらの多項目のデータを効率よく，メモリに格納し，取り扱うにはデータを配列として取り扱うようにします．

すなわち，1次元の表や2次元の表を操作することになります．

ここでは，表の作成，表の中へのデータの格納，表からのデータの取り出しについて理解します．

1 1次元の配列表の作り方

1月から6月までの各月の収入金額を1つの変数名でメモリの中へ格納しておくには，次のようにします．

プログラム4-1

```
10   DIM A(6)
20   READ A(1), A(2), A(3), A(4), A(5), A(6)
30   DATA 1000, 2000, 3000, 4000, 5000, 6000
40   PRINT A(1) ; A(2) ; A(3) ; A(4) ; A(5) ; A(6)
1000  END
```

このプログラムによれば，下表のように各月のデータが格納されます．

図1

変数名	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)
月	1	2	3	4	5	6
収入金額	1000	2000	3000	4000	5000	6000

行番号10のDIM文はメモリにデータをまとめて格納する領域を確保するときに使います．これを**配列**といいますが，A(6)は**配列変数**と呼ばれ，ここでは，A(0)～A(6)までに7つのデータを格納できます．

プログラム4-1では，A(0)は使用していません．

行番号40では，READ, DATA文で読み込まれたデータを確認する意味で出力しました．

〔演習〕

プログラム4-1を改造して，1月～6月の任意月をキー入力するとその月の収入金額が表示されるプログラムを作り，実験してみましょう．

```
50 INPUT "ナンガツ デスカ" ; M
60 PRINT M ; "月ノシュウニウハ" ; A(M) ; "円デス"
70 PRINT
80 GOTO 50
```

以上の行番号50～70を追加してRUNします．

〔演習〕

配列変数N＄(5)に商品名を5種類，格納し，商品コードをキー入力したら商品名が表示されるプログラムを作りましょう．

表2

変数名	N＄(1)	N＄(2)	N＄(3)	N＄(4)	N＄(5)
商品コード	1	2	3	4	5
商品名	イチゴ	リンゴ	ミカン	スイカ	ブドウ

プログラム4-2

```
100  DIM N$(5)
110  READ N$(1), N$(2), N$(3), N$(4), N$(5)
120  DATA イチゴ, リンゴ, ミカン, スイカ, ブドウ
130  PRINT N$(1) ; N$(2) ; N$(3) ; N$(4) ; N$(5)
140  INPUT "コードハ" ; C
150  PRINT "コード" ; C ; "ハ ヒンメイ" ; N$(C) ; "デス"
160  PRINT
170  GOTO 140
```

文字データは引用符〝 〟でデータを囲います．また，変数は文字型を示す＄記号を使います．

RESTORE文は，READ文が実行されるときはDATA文の最初から読み込んでくることを指示するものです．

〔注意〕

配列変数　A(1), A(2), A(3)………は
数値変数　A1，A2，A3とは意味が違います．すなわち配列変数では（ ）の中も変数になりえます（**添字**という）．添字が1個の配列を**1次元の配列**といいます．

[2] 2次元の配列表の作り方

1月から3月までの各月の収入，支出をメモリの中へ格納するには，次のように2次元の配列を使うことになります．

表3

項目＼月	1	2	3
収入	1000	2000	3000
支出	100	200	300

たとえば，
B(2，3)=300
B(1，2)=2000
を意味します．

表4

0	1	2	3
B(0，0)	B(0，1)	B(0，2)	B(0，3)
B(1，0)	B(1，1)	B(1，2)	B(1，3)
B(2，0)	B(2，1)	B(2，2)	B(2，3)

DIM B(2，3)は，左のように2次元の表をメモリの中に作ります．

すなわち，配列変数B(2，3)で，2個の添字のうち，右の添字で月を，左の添字で項目を示します．

残高は，1月の分をB(0，1)，2月分をB(0，2)，3月分をB(0，3)に計算して格納します．

プログラム4-3

```
10  DIM B(2,3)
20  READ B(1,1), B(2,1)
30  READ B(1,2), B(2,2)
40  READ B(1,3), B(2,3)
50  DATA 1000, 100
60  DATA 2000, 200
70  DATA 3000, 300
80  RESTORE
150 INPUT "ナンガツ デスカ" ; M
160 PRINT M ; "月"
170 PRINT "シュウニュウ ハ" ; B(1,M) ; "円"
180 PRINT "シシュツ ハ" ; B(2,M) ; "円"
190 B(0,M)=B(1,M)-B(2,M)
200 PRINT "ザンダカ ハ" ; B(0,M) ; "円"
210 PRINT
220 GOTO 150
1000 END
```

〔演習〕

プログラム4-3のうち，各月の収入，支出金額をキーボードより入力できるようにします．

プログラム4-4

```
10  DIM B(2,3)
20  INPUT "ナンガツデスカ" ; M
30  INPUT "シュウニウ ハ" ; B(1, M)
40  INPUT "シシュツ ハ" ; B(2,M)
50  B(0, M)=B(1, M)-B(2, M)
60  PRINT M ; B(1, M),; B(2,M) ; B(0, M)
70  PRINT : GOTO 20
1000 END
```

〔演習〕

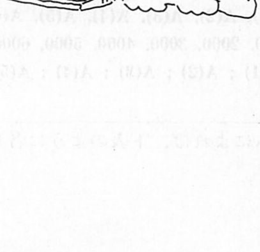
次のプログラムを格納して，数値型と文字型の配列変数の取り扱いについて確かめてください．

プログラム4-5

```
100  DIM N$(3), T(3)
110  READ N$(1), T(1), N$(2), T(2), N$(3), T(3)
120  DATA メロン, 800, スイカ, 500, ブドウ, 350
130  PRINT N$(1), T(1)
140  PRINT N$(2), T(2)
150  PRINT N$(3), T(3)
160  INPUT "コード ハ 1-3" ; I
170  PRINT I ; N$(I) ; "タンカ" ; T(I) ; "円"
180  PRINT : PESTORE
190  GOTO 160
1000 END
```

このように配列変数はひとつの変数名で添字を付けることによって数多くのデータを取り扱うことができます．毎月の電気料金，ガス料金，電話料金などの一覧表や，生徒ごとのテスト結果の集計表など，実用プログラムでは表の使用が必ずあります．

配列変数を使ったプログラムでは，実際には，ここで述べたような**READ, DATA**文の形ではなく，繰り返し実行文を使ったものになり，ずっと短いプログラムになります．これについては，次の**FOR～NEXT**文で説明します．



5 第5日 繰り返し処理のプログラム

〔目標〕

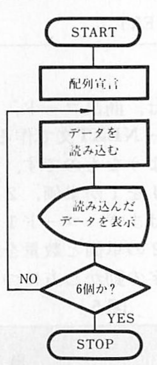
前日の配列のところでREAD, DATA文により，何度もデータを読み込むところがありました．プログラムを設計するときひとつの文を繰り返し何度も実行するようなことが多いことに気がつきます．繰り返しの回数があらかじめ確定する場合には，FOR～NEXTという文を使ってプログラムを作ると便利です．また，プログラムのステップ数も少なくなります．

ここでは，繰り返し処理のプログラム，すなわち，ループのあるプログラムの作り方について，『単一ループ』と『多重ループ』を例に理解します．そして，まとめとして，『仕入作業のプログラム化』と『仕入ファイルの作成プログラム例』について研究します．

1 単一ループのプログラムの作り方

プログラム4-1でループを使って改良すると，図1のようになります．６個のデータを配列変数A(6)の中へ格納するには添字が1～6まで変ることに注意しましょう．そして，添字の増分単位は1であることを理解します．ある文を繰り返し，繰り返し実行させるとき，その繰り返し回数があらかじめ確定する場合には，次のような形式でループを作ります．

図1



FOR 制御変数＝初期値 **TO** 最終値 **STEP** 増分
この間に，繰り返し実行される文を書く．
NEXT 制御変数

このFOR～NEXT文を使って，**プログラム4-1**を改良したものが，次の**プログラム5-1**です．実際にプログラムをキー入力して実行してみてください．

〔演習〕

プログラム5-1

```
10   DIM A(6)
20   FOR I=1 TO 6 STEP 1
30   READ A(I)
40   PRINT A(I) ;
50   NEXT I
60   DATA 1000, 2000, 3000, 4000, 5000, 6000
70   RESTORE
1000 END
```

このプログラムでは，配列変数の添字が変数Iになり，その変数Iを1からループを廻るごとに1ずつ増分し，6になるまでFOR文とNEXT文で囲まれた文を繰り返し実行します．行番号30，40のREAD文，PRINT文はそれぞれ6回実行されます．

添字変数Iが1から6まで変っていく様子は次のプログラムで確めてみましょう．

〔演習〕

プログラム5-2

```
100  FOR I=1 TO 6 STEP 1
110  PRINT I ;
120  NEXT I
1000 END
```

プログラム5-1が格納されている場合には**RUN100**として実行します．行番号100の**STEP 1**は増分が＋1のときにのみ省略できます．省略されている場合は増分は＋1となります．

1回目の実行では**PRINT I**文で1が表示されます．**NEXT I**文でIが＋1されI＝2となり，再び**PRINT I**が実行され，2が表示されます．次の**NEXT I**文でさらにIが＋1され3となり，以下6になるまで繰り返します．この様子を図2にまとめておきました．

PC-8001での**FOR～NEXT**文ではどんな制御変数の値でも，1回は**FOR～NEXT**文が実行されます．

図2　制御変数の増分の様子

制御変数	表示内容	増　分
I	PRINT I	I=I+1
1	1	2
2	2	3
3	3	4
4	4	5
5	5	6
6	6	7

〔演習〕

次のプログラムを制御変数の値と増分に注意して実験的に確かめてください．

プログラム5-3

```
150  FOR I=10 TO 1 STEP -1
160  PRINT I ;
170  NEXT I
1000 END
```

プログラム5-4

```
200  FOR I=1 TO 20 STEP 2
210  PRINT I ;
220  NEXT I
1000 END
```

表示結果をよく見て制御変数Iの変化する様子を理解しましょう．

〔演習〕

10個のデータをキーボードから配列変数DAT(10)に格納するとともに総和を求めるプログラムを作りましょう．

プログラム5-5

```
250  DIM DAT(10)
260  FOR I=1 TO 10
270  INPUT "データ オ イレヨ"; DAT(I)
280  DAT(0)=DAT(0)+DAT(I)
290  NEXT I
300  PRINT "10コノデータ ハ ツギノ トオリ デス"
310  FOR I=1 TO 10
320  PRINT  "DAT(";I;")=";DAT(I)
330  NEXT I
340  PRINT
350  PRINT "データ ノ ゴウケイ ハ"; DAT(0);"デス"
1000 END
```

〔演習〕

10種類の商品名を商品コードとともにメモリに格納するプログラムを**プログラム5-1**を参照して作ります．

プログラム5-6

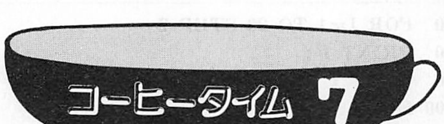
```
400  DIM N$(10)
410  FOR I=1 TO 10
420  PEAD N$(I)
430  PRINT N$(I) ;
440  NEXT I
450  RESTORE
460  DATA "イチゴ","ブドウ","ミカン","リンゴ","バナナ"
470  DATA "レモン","メロン","スイカ","モモ","ナシ"
1000 END
```

ここで，Ｉは商品コードの役目をしています．

商品名がメモリに格納された様子を図3に示す．**DATA**文での文字型のデータは6桁とするため，6桁に満たないモモ，ナシ，メロンなどは，空白を後に入れて引用符で囲んでおくと表示するときに便利です．

図3

変数名	コード	品　名
N$(0)	0	
N$(1)	1	イチゴ
N$(2)	2	ブドウ
N$(3)	3	ミカン
N$(4)	4	リンゴ
N$(5)	5	バナナ
N$(6)	6	レモン
N$(7)	7	メロン
N$(8)	8	スイカ
N$(9)	9	モ　モ
N$(10)	10	ナ　シ



プログラム5-7

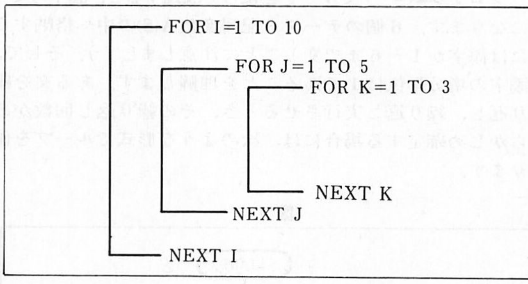
```
500  CONSOLE,,,1
510  C=RND(1)*7+1
520  CC=INT(C)
530  COLOR CC, 233, 0
540  INPUT "1-40マデ ノ ドレカ"; N
550  FOR I=1 TO N
560  PRINT CHR$(135);
570  NEXT I
580  PRINT
590  GOTO 510
1000 END
```

このプログラムはキー入力された数だけ■記号で横に棒グラフを表示するものです．行番号560の**PRINT**文で**CHR$(135)**の代わりに引用符を使ってグラフィック記号■を囲んで**560 PRINT "■"**；としても，同じことです．

２多重ループのあるプログラムの作り方

FOR～NEXT文はループの中にもまた**FOR～NEXT**文がくる場合があります．この様子は図4に示すごとくですが，このようなプログラムの例について実験してみましょう．

図4



〔演習〕

このプログラムは，商品コード，単価，数量表を作るプログラムで，**FOR～NEXT**文で作るものです．

表は図5に示すようなものです．配列変数を**T(I, J)**としＩを商品コード，Ｊを1が単価，2が数量とします．データを読み込む順序はまず，コード1の単価，数量を読み込み，次に，コード2の単価と数量を読み込むようにします．この配列要素と添字の動かし方について図6に示しました．

図5

項目／コード		単　価	数　量
0	0，0	0，1	0，2
1	1，0	1，1	1，2
2	2，0	2，1	2，2
3	3，0	3，1	3，2

図6

```
T(1, 1)= 50  T(1, 2)=500
T(2, 1)=100  T(2, 2)=300
T(3, 1)=200  T(3, 2)=150
I=1     I=2     I=3
J=1, 2  J=1, 2  J=1, 2
```

プログラム5-8

```
10  DIM T(3, 2)
20  FOR I=1 TO 3
30  FOR J=1 TO 2
40  READ T(I, J)
50  PRINT "T(" ; I ; " , " ; J ; ")=" ; T(I, J)
60  NEXT J
70  NEXT I
80  DATA 50, 500, 100, 300, 200, 150
90  RESTORE
1000 END
```

このプログラムでは2重のループとなりました．すなわち，行番号20～70と30～60の範囲です．DATA文は，単価，数量，単価，数量の順序で並べておく必要があります．

〔演習〕

九九表を作るプログラムを考えてみましょう．

プログラム5-9

```
100  FOR I=1 TO 9
110  FOR J=1 TO 9
120  PRINT I ; "×" ; J "=" ; I*J
130  NEXT J : PRINT
140  NEXT I
1000 END
```

結果は図7のようになります．

〔演習〕

10品目程度の商品を例として商品の仕入作業を行ない，仕入ファイルを作るプログラムを考えてみましょう．

仕入の作業設定は次の通りです．

> 商品コード(1～10)N，仕入単価ST(N)，仕入数量Sを入力し，キー入力作業の正誤を確認して，仕入金額Y♯を計算し，また，仕入総数量SS♯(N)，仕入総金額SD♯(N)について，商品ごとに計算し，仕入ファイル一覧表を作成する．全商品の仕入総金額SZ♯を算出する．

出力結果例を図8に示します．

プログラム5-10

```
10  DIM  ST(10), SS♯($0), SD♯(10) ; N$(10)
40  SZ♯=0
```

図7

```
1×1=1     9×1=9
1×2=2     9×2=18
1×3=3     9×3=27
1×4=4     9×4=36
1×5=5     9×5=45
1×6=6     9×6=54
1×7=7     9×7=63
1×8=8     9×8=72
1×9=9     9×9=81
```

図8

＊＊＊ シイレ ファイル イチラン ヒョウ ＊＊＊

コード	ヒンメイ	スウリョウ	キンガク
1	イチゴ	100	1,000
2	ブドウ	200	4,000
3	ミカン	300	9,000
4	リンゴ	0	0
5	バナナ	0	0
6	レモン	0	0
7	メロン	0	0
8	スイカ	100	1,000
9	モモ	300	6,000
10	ナシ	0	0
			¥21,000

プログラム・リスト

```
1500  CONSOLE , , , 0
1510  FOR I=1 TO 10
1520  READ N$(I)
1530  NEXT I : RESTORE
1540  DATA "イチゴ"
1550  DATA "ブドウ"
1560  DATA "ミカン"
1570  DATA "リンゴ"
1580  DATA "バナナ"
1590  DATA "レモン"
1600  DATA "メロン"
1610  DATA "スイカ"
1620  DATA "モモ"
1630  DATA "ナシ"
1640  PRINT "シイレ ファイル サクセイ"
1650  PRINT
1660  INPUT "コード" ; N
1670  IF N=999 GOTO 2000
1680  INPUT "シイレ タンカ" ; ST(N)
1690  INPUT "シイレ スウリョウ" ; S
1700  COLOR 4
1710  PRINT N ; "  " ; N$(N) ; ST(N) ; S
1720  COLOR 0
1730  INPUT "y/n" ; A$
1740  IF A$="n" GOTO 1660
1750  Y♯=ST(N)*S
1760  SS♯(N)=SS♯(N)+S
1770  SD♯(N)=SD♯(N)+Y♯
1780  GOTO 1660
2000  REM ＊＊＊ シイレ ファイル ＊＊＊
2010  LPRINT : LPRINT "＊＊＊ シイレ ファイル イチラン ヒョウ ＊＊＊"
2020  LPRINT
2040  LPRINT "コード ヒンメイ スウリョウ キンガク"
2050  LPRINT
2060  FOR X=1 TO 10
2070  LPRINT USING "♯♯" ; X ;
2070  LPRINT N$(X) ;
2080  LPRINT USING "♯♯♯♯  ♯♯♯♯♯♯♯♯♯, " ; SS♯(X), SD♯(X)
2090  SZ♯=SZ♯+SD♯(X)
2100  NEXT X
2110  LPRINT "____________________________________"
2120  LPRINT SPC(17) ;
2130  LPRINT USING "¥¥♯♯♯♯♯♯♯♯♯," ; SZ♯
2140  INPUT "OK" ; A$
2150  GOTO 1640
3000  END
```



6 第6日 マイコンに比較・判断をさせる

〔目標〕

計算結果や入力したデータがある値より大きいか小さいか，また，ある範囲内にあるのかないのかを調べ，その結果それぞれ該当する処理をするプログラムへ分岐していくような場合はたくさんあります．

『もし～ならば――する』というような文は，マイコンではIF文という比較・判断命令があります．

このIF文をうまく使いこなすと，あたかも，マイコンが複雑かつ高機能な判断をしてくれると錯覚してしまうほど素晴らしいプログラムを作ることが可能です．

ここでは，数値の大，小や，文字の判断ができるプログラムの作り方について学習し，IF文を理解します．

1 数値の判断

キー入力した数値が100より大きいか，小さいかを判断するプログラムを作って実験してみる．

プログラム6-1

```
10  INPUT "x="; X
20  IF X>=100 GOTO 50
30  PRINT "チイサイョ"
40  GOTO 10
50  PRINT "オオキイョ"
60  GOTO 10
```

20 1F X＞＝100 GOTO 50の文は，Xの値が100に等しいかそれ以上ならば，行番号50へジャンプして，PRINT"オオキイヨ"を実行する．100未満の場合はこの文の次の文，すなわち行番号30のPRINT"チイサイヨ"を実行する．

この文の型は，IF論理式GOTO行番号となっている．論理式で使える関係演算子として，

＝…等しい
＜＞，＞＜…等しくない
＜…小さい
＞…大きい
＜＝，＝＜…等しいか小さい
＞＝，＝＞…等しいか大きい

などがあります．

〔演習〕

0から7までの数値を受け付けるプログラムを作りましょう．それ以外の数値はプリントしません．

プログラム6-2

```
100  INPUT "スウジ オ イレヨ"; X
120  IF X<0 GOTO 100
130  IF X>7 GOTO 100
140  PRINT "カズ ハ"; X; "デス"
150  GOTO 100
```

ある範囲の数値しかキー入力で受け付けないようにするには上のようなプログラムとします．

〔演習〕

[0]～[9]までのキー入力しか受け付けないようなプログラムを作るには**プログラム6-2**のどこを改良すればよろしいでしょう．

〔演習〕

数あてゲームのプログラムを作ってみましょう．当てる数は最初に前もってプログラムのなかで代入文で定義しておきます．

プログラム6-3

```
200  A=135
210  INPUT "カズオアテテクダサイ"; X
220  IF X<135 GOTO 260
230  IF X>135 GOTO 270
240  IF X=135 GOTO 280
250  GOTO 210
260  PRINT "チイサスギマス": GOTO 210
270  PRINT "オオキスギマス": GOTO 210
280  PRINT "アタリマシタ"
290  END
```

2 文字の判断

アルファベット・キーA～Zだけを受け付けるようなプログラムはよくあります．たとえば，2者択一の場合，YESのときは[Y]のキーを，NOのときは[N]のキーを入力させるときがあります．このようなときは[Y]のキーを押したか[N]のキーを押したか調べ判断しなければなりません．

文字を判断するときは次のようなプログラムになります．

〔例〕

プログラム6-4

```
300  INPUT "YESナラ(Y), NOナラ(N) オイレヨ"; A$
310  IF A$="Y" GOTO 340
320  1F A$="N" GOTO 350
330  GOTO 300
340  PRINT "YES": GOTO 360
350  PRINT "NO"
360  PRINT: GOTO 300
```

入力文字の判断や文字列の判断にはIF文の論理式は，判断すべき文字列を引用符“ ”で書き，関係演算子で文字変数と結びます．

〔演習〕

アルファベット・キーのA～Zまでを受け付け，他のキー入力は受け付けないようなプログラムを作ってください．

〔演習〕

商品名，メロンをキー入力したら商品コード7とメロンを印刷するプログラムを作ってください．それ以外のキー入力は受け付けないものとします．

次のプログラムを比べて，IF文と先に学習したFOR～NEXT文の相違点について考えてみてください．プログラムは2^nを計算するプログラムです．結果を図1に示します．

プログラム6-5

```
10  N=1: I=1
20  N=N*2
30  PRINT "2^"; I; "="; N
40  I=I+1
50  IF I<=16 THEN GOTO 20
60  END
```

プログラム 6-6

```
10  N=1
20  FOR I=1 TO 16
30  N=N*2
40  PRINT "2^"; I; "="; N
50  NEXT I
60  END
```

図1　結果

```
2^1 =2
2^2 =4
2^3 =8
2^4 =16
2^5 =32
2^6 =64
2^7 =128
2^8 =256
2^9 =512
2^10=1024
2^11=2048
2^12=4096
2^13=8192
2^14=16384
2^15=32768
2^16=65536
```

3 IF〜THEN---ELSE

『もし〜ならば----をする．さもなければ——をする』というような論理をすすめときにはIF〜THEN----ELSE—— 文を使います．

プログラム 6-1 の数値の大小を判断するプログラムをこの形で書いてみます．

〔例〕

プログラム 6-7

```
410  A=100
420  INPUT "x="; X
430  IF X>=A THEN M$="オオキイョ" ELSE M$="チイサイョ"
440  PRINT M$
450  GOTO 420
460  END
```

PRINT文と**GOTO**文の数が1つずつ合計2行減少しました．条件が成立したとき（ここでは，XがAに等しいか，またはそれ以上のとき）は**THEN**の次の文を実行し，条件が成立しないときは**ELSE**以下の文が実行されます．

〔例〕

プログラム 6-2 をIF〜THEN------ELSE———文で書き直すと次のようなプログラムになります．

プログラム 6-8

```
500  INPUT "スウジ1-7オイレヨ"; X
510  1F X<0 OR X>7 THEN 500 ELSE PRINT
     "カズ ハ"; X ;"デス"
520  PRINT : GOTO 500
```

論理式 **X<0 OR X>7** は，2つの式が論理演算子ＯＲで結ばれており，**または**の意味となります．このほかに論理演算子としては，NOT, AND, XOR, IMP, EQUなどがありますが，これらについては後述します．**THEN**の後にはこのように行番号だけを書くこともできます．**ELSE**の後も同様に行番号だけでも書けます．

4 条件が多い場合のIF文による処理

IF文では，**THEN**や**ELSE**にさらに**IF**文を書くことができます．そのため，複雑な条件判断を行なうときは多重の**IF**文によって処理するプログラムを作ります．

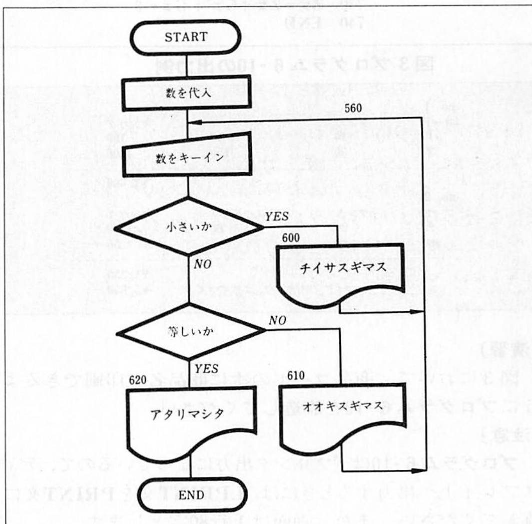
〔例〕

プログラム 6-3 には**IF**文が3つもあります．これは，条件としてX<135，X>135，X=135の3つがあることになります．このプログラムは次のように書くこともできます．フロチャートを図2に示します．

プログラム 6-9

```
550  A=135
560  INPUT "カズ オ アテテ クダサイ"; X
570  IF X<135 THEN 600 ELSE IF X=135 THEN
     620 ELSE 610
580  GOTO 560
600  PRINT "チイサスギマス"; GOTO 560
610  PRINT "オオキスギマス"; GOTO 560
620  PRINT "アタリマシタ"
630  END
```

図2



〔例題〕

売上伝票の発行

取扱商品数10種類の売上伝票発行プログラムを考えてみる．伝票1枚に取引商品件数は5件までとする．売上データとしては商品コード，単価，数量をキー入力する．伝票は何枚でも発行できるものとし，伝票No.を印刷するものとする．

発行する伝票の印刷項目は商品コード，単価，数量，金額とし，伝票1枚ごとの小計を計算して印刷するとともに数枚の伝票にわたる同一取引に対しては各伝票の総合計を計算して印刷する．

また，売上ファイルを作成するために，各商品ごとの売上総数量，売上総金額の配列を用意し，伝票発行の時点で積算していくものとする．

ここでは，小計の指示，合計の指示を＊キー，￥キーを入力することで行ない，**IF**文を使って判断するようにプログラムを作ります．プログラム例を次に示します．

プログラム 6-10

```
10   DIM N(5), T(5), S(5), Y#(5)
20   DIM S#(10), D#(10)
30   G#=0 : P=1
500  T#=0 : I=0                                  売上伝票1枚毎の小計T#=0，商品件数I
510  I=I+1
520  INPUT "ヒンメイ ハ"; N(I)                    商品コード入力
530  INPUT "タンカ ハ"; T(I)                      単価入力
540  INPUT "スウリョウ ハ"; S(I)                  数量入力
550  Y#(I)=T(I)*S(I)                             金額計算
560  N=N(I)                                      Nに商品コード
570  S#(N)=S#(N)+S(I)                          } 各商品毎の売上総数量，総売上金額
580  D#(N)=D#(N)+Y#(I)                         }
585  IF I=5 GOTO 610                             商品件数5件なら小計
590  INPUT "ショウケイ ナラ(*)ヲ イレル"; A$        小計なら*をキー入力
600  IF A$< >"*" GOTO 510                        小計でなければ，次の商品を入力する
610  LPRINT "no."; P:PRINT                       伝票No.印刷
615  LPRINT "コード"; SPC(10); "タンカ"; SPC(6) ; "スウリョウ" ; SPC(10) ; "キンガク"   目出し印刷
620  FOR N=1 TO I
630  LPRINT USING "###        ####          ####          #########";
     ; N(N), T(N), S(N), Y#(N)
640  T#=T#+Y#(N)
650  NEXT N
660  LPRINT "--------------------------------------------------------------------"
670  LPRINT TAB(33) : LPRINT USING "¥¥#########,,"; T#
680  G#=G#+T#: P=P+1                             頁数更新，1人分の総合計
690  INPUT "ゴウケイ ナラ(¥)ヲ イレル"; A$          1人分合計なら¥入力
700  IF A$="¥" GOTO 720
710  GOTO 500
720  LPRINT SPC(14) : LPRINT "マイド アリガトウ ゴザイマス"; TAB(25) ;
725  LPRINT USING "¥¥#########,,"; G#
730  Z#=Z#+G# : G#=0
740  END
```

図3 プログラム 6-10の出力例

```
no. 1
コード      タンカ        スウリョウ       キンガク
  1          10             10             100
  3          30            150            4500
-------------------------------------------------
                                        ¥4,600
no. 2
コード      タンカ        スウリョウ       キンガク
  9          20             60            1200
  8          10              5              50
-------------------------------------------------
                                        ¥1,250
            マイド アリガトウ ゴザイマス    ¥5,850
```

〔演習〕

図3において，商品コードの次に商品名を印刷できるようにプログラム6-10を改造してください．

〔注意〕

プログラム6-10は，プリンタ出力になっているので，ディスプレイ上へ出力するときには，LPRINT文をPRINT文に直してください．また，画面は1行80字とします．

コーヒータイム 8

次のプログラムを実行して，楽しんでください．正確な判断をする式ではないので念のため申し添えます．

プログラム 6-11

```
700  INPUT "アナタノ ネンレイ ハ"; A
710  INPUT "アナタノ シンチョウ ハ"; B
720  INPUT "アナタノ タイジュウ ハ"; C
730  INPUT "アナタノ ケツアツ ハ"; D
740  PRINT
750  C$="アナタノ タイジュウ ハ セイジョウデス"
760  D$="アナタノ ケツアツ ハ セイジョウデス"
770  E$="アナタ ハ ヒマン デス"
780  F$="アナタノ ケツアツ ハ タカスギ デス"
790  IF C>(B-100)*0.9 THEN PRINT E$ ELSE PRINT C$
800  IF D>120+(A-20)/2 THEN PRINT F$ ELSE PRINT D$
1000 END
```

7 第7日 算術演算と論理演算のプログラム

〔目標〕

四則演算やベキ乗計算，分数計算などの数値計算から，大小の比較演算，論理演算，三角関数や対数関数などの関数計算などが日常計算ではよく使われています．ここでは，これらの初等演算について，整数，固定小数点，浮動小数点の場合ならびに6桁表示の単精度計算，16桁表示の倍精度計算についてダイレクト・モードによって各命令を実行し，実験的にその命令の機能について学習します．

また，売上一覧表の作成と在庫ファイルの作成プログラムを通じて，横の計算と縦の計算の実例を理解し，配列を使った計算と印刷のプログラム作成技法を理解します．

1 算術演算

❶四則演算（＋，－，×，÷）

足し算，引き算，掛け算，割り算をダイレクト・モードで実験してみよう．

〔例1〕 **print 25＋3000** RET

3025

答の3025は符号の＋が表示されていません．

〔例2〕 **?230－500** RET

－270

ここで，？はPRINTの省略形です．

〔例 3〕 **?35＊10** RET

掛け算の記号×はマイコンでは＊(アスタリスク)になります．

〔例 4〕 **?1/3** RET

.333333

割り算の記号÷はマイコンでは／(スラッシュ)になります．

❷ベキ乗の演算 (X^n)

ベキ計算の記号としては＾を使います．

〔例 5〕 **?2＾8** RET

256

以上は単精度といって，6桁まで表示しますが，もっと大きな数を取り扱うには倍精度といって，16桁まで表示できる記号＃を数字の最後に付けます．

❸倍精度計算（16桁）

〔例 6〕 **?1＃／3＃** RET

.3333 3333 3333 3333

❹カッコのついた計算

3×(8－2)は，まずカッコ内の計算8－2を行ない，その結果に3が掛けられます．

〔例 7〕 **?3＊(8−2)** RET

18

❺分数の計算

$\frac{1}{3}+\frac{2}{5}$のような分数の計算は次のようになります．割り算が先にそれぞれ行なわれ，その後足し算が行なわれます．

〔例 8〕 **?1/2+3/5** RET

1.1

❻演算の優先順位

計算の順序は，まず（　）の中の計算が行なわれ，その後，ベキ乗，掛け算か割り算，足し算か引き算の順序です．同じ順位のときは左から順々に行なわれます．

〔例 9〕 **?2＊(5+3)/2−2＊(7−5)**

4

（5＋3)＝8，(7－5)＝2となり，2×8＝16，16÷2＝8，2×2＝4となって，最終的に8－4＝4となります．

❼整数の計算

整数形式の定数や変数の取り得る値は－32,768から＋32,767までのすべての整数であり，整数はコンマや小数点を含みません．整数型変数の型宣言文字は％です．

〔例10〕 **A％＝45.36 : ?A％** RET

45

〔例11〕

整数の除算は￥記号を使います．その商は整数に丸められます． **?15￥4** RET

3

〔例12〕

整数の剰余計算は除算の剰余が整数で得られます．演算記号としてMODを使います．

?17.4 mod 5 RET

2

❽固定小数点の計算

〔例13〕 **?4.63＊8.15** RET

37.7345

〔例14〕 **?6.28/2.31** RET

2.71861

〔例15〕 **?8.2653748593/9.2** RET

.8984103294193665

〔例16〕 **?7.2＊0.384621538** RET

2.769275000239264

例13，14では結果は6桁で表示されるが，例15，16のように6桁以上の数を入力すると結果は自動的に16桁表示されます．

❾浮動小数点の計算

指数形式で表現された$2.323\times10^7\times7.25\times10^{-2}$というような計算です．マイコンでは指数部分をEで表現します．その範囲は－38から＋38までです．

〔例17〕 **?2.323E7＊7.25E−2** RET

1.68418E＋06

ここで，EはExponential（指数）のことで10のベキ乗を意味します．2.323や7.25の部分は仮数部といいます．答は1.68418×10^6を意味します．

倍精度の浮動小数点形式の計算にはEのかわりにDを使います．

〔例18〕 倍精度の浮動小数点の計算

?1.258694235D8＊3.568143621D10 RET

4.491201805404725D＋18

まとめ 数値定数と数値変数

数値定数	例
整数	32767
固定小数点	25.3
浮動小数点	103E5
10進	100
16進	&H64
8進	&144(&O144)

まとめ 変数と形宣言文字

変　数	形宣言文字	例
整数変数	％	DATA ％
単精度変数	！	DATA ！
倍精度変数	＃	DATA ＃

単に，DATAという形宣言文字のない数値変数は単精度変数とみなします．

ここで，単精度とは有効桁7桁以下の数，または，Eを使った指数形式表現，最後に感嘆符！を付けたものを示します．

一方，倍精度とは，有効桁8桁以上（17桁まで）の数，または，Dを使って指数表現されたもの，最後にナンバー記号＃をつけたものを示します．

〔注意〕

桁数が7桁を起える大きな数になるときは倍精度変数を使って計算したりします．整数変数を使うと小数点以下は丸められて計算されるので注意してください．

〔演習〕

32,767円以上のお金は倍精度計算が必要となります．次のプログラムを実験してみてください．

プログラム7-1

```
10 INPUT "a＃="; A＃
20 PRINT USING "￥￥＃＃＃＃＃＃＃＃,,"; A＃
30 PRINT
40 GOTO 10
```

〔例〕

¥記号による整数の除算

?15.6¥4 はいくらになるか……………………………(4)

〔例〕 整数の剰余計算

〔例〕 ? 27.8 mod 5 はいくらになるか……(3)

〔例〕 a＃＝5＃/3 ：? a＃ はいくらか……
(1.666666666666667)

〔例〕 a＝5＃/3 ：? a はいくらか…(1.66667)

〔例〕 b％＝657.842 ：? b％ はいくらか…(658)

〔例〕 x＝1.05 y＃＝x ：? x, y＃ はいくらか…
(1.05 1.049999952316284)

2 関係演算

2つの値を比較するときに使われる演算で演算子としては2つの値が等しいか，2つの値の大小比較などを行なうものがあります．比較の結果は〝真〟ならば－1，〝偽〟ならば0になります．IF文の中で条件式として使い，プログラムの流れを決定するのによく使われます．

❶大，小の比較

〔例19〕

?9>5 RET

－1

9は5より大きいことは正しいことを示し，その値は，〝真〟(－1)です．

〔例20〕

?9<5 RET

0

9は5より小さいことは間違いであり，その値は，〝偽〟(0)です．

❷等しいことの比較

〔例21〕

?9=5, 9=9

0 －1

9＝5は〝偽〟(0)で，9＝9は〝真〟(－1)であることを示す．

9＝5は〝偽〟(0)で，9＝9は〝真〟(－1)であることを示す．

❸その他の関係演算子

< >，> <は等しくない，<＝，＝<等しいか小さい，>＝，＝>は等しいか大きいを意味します．

〔演習〕 **プログラム7-2**

```
10 INPUT A
20 PRINT "< >"; A< >100
30 PRINT "<="; A<=100
40 PRINT ">="; A>=100
50 GOTO 10
```

上のプログラムを実行し，Aの値を50，110，100などを与えて，結果を比較すると，50の場合－1，－1，0，110の場合－1，0，－1，100の場合0，－1，－1となることを確かめてください．すなわち，条件が成り立ったときは－1，成り立たないときは0という値をとることがわかります．

〔例題〕

売上一覧表の作成

下の図に示すような売上伝票が発行され，売上商品のコードI，数量S＃(I)，金額D＃(I)が判っており，すでに配列S＃(I)，D＃(I)にデータが格納されている状態を想定して売上一覧表の作成プログラムを作る．取扱商品コードは1～10まであるとする．

図1 売上伝票

```
no. 1
コード      タンカ     スウリョウ      キンガク
  1          10          20            200
-----------------------------------------------
                                      ¥200

no. 2
コード      タンカ     スウリョウ      キンガク
  8          250          5            1250
-----------------------------------------------
                                    ¥1,250
             マイド アリガトウ ゴザイマス  ¥1,450
```

プログラム例を図2に示す．ここでZ＃は売上総合計金額である．S＃(I)，D＃(I)，Z＃の計算式は，前述した第6日目の**プログラム6-10**を参照してください．図3にその出力結果を示す．

図2 売上一覧表プログラム

```
1000 LPRINT : LPRINT "*** ウリアゲ イチラン ヒョウ ***"; LPRINT
1005 LPRINT : LPRINT "コード スウリョウ キンガク" : LPRINT
1010 FOR I=0 TO 10
1020 LPRINT USING "### #####  #########"; I, S#(I), D#(I)
1030 NEXT I
1040 LPRINT "------------------------------"
1050 LPRINT SPC(10) ;
1060 LPRINT USING "¥¥########,,"; Z#
1070 INPUT "スミマシタラ キーイン", B$
1080 END
```

図3 出力結果

```
*** ウリアゲ イチラン ヒョウ ***

コード   スウリョウ   キンガク
  0         0           0
  1        20         200
  2         0           0
  3         0           0
  4         0           0
  5         0           0
  6         0           0
  7         0           0
  8         5        1250
  9         0           0
 10         0           0
-------------------------------
                    ¥1,450
```

〔例題〕 **在庫一覧表の作成**

1 **演習1**

『5日目』の1演習1で作った仕入ファイル一覧表から売上後の在庫ファイルを作るプログラムを作成する．

在庫数量ZS＃(J)＝仕入数量SS＃(J)－売上数量S＃(J)
在庫金額ZD＃(J)＝在庫数量ZS＃(J)＊仕入単価ST(J)
総在庫金額ZZ＃＝仕入総金額SZ＃－売上総金額Z＃

とします．そして，これらのデータのうち，SS＃(J)，S＃(J)，ST(J)，商品名N＄(J)，SZ＃，Z＃はすでに他のプログラムの実行によりその値が確保されているものとします．

『5日目』の『仕入ファイル作成プログラム』，『6日目』の『売上伝票作成プログラム』，『7日目』の『売上ファイル作成プログラム』を参照してください．

在庫一覧表を作成するプログラム例を図4に示します．また，その出力結果例を図5に示します．

図5 出力結果

*** ザイコ イチラン ヒョウ ***

コード	ヒンメイ	タンカ	スウリョウ	キンガク
1	イチゴ	10	80	800
2	ブドウ	20	100	2,000
3	ミカン	30	100	3,000
4	リンゴ	40	400	16,000
5	バナナ	500	55	27,500
6	レモン	100	50	5,000
7	メロン	200	50	10,000
8	スイカ	250	35	8,750
9	モモ	84	25	2,100
10	ナシ	150	35	5,250
				¥80,400

図4　在庫一覧表作成プログラム

```
2500  REM ＊＊＊ ザイコ イチラン ヒョウ サクセイ ＊＊＊
2510  PRINT CHR$(12)
2520  LPRINT "＊＊＊ ザイコ イチラン ヒョウ ＊＊＊"
2530  LPRINT
2540  LPRINT "コード ヒンメイ タンカ スウリョウ       キンガク"
2550  LPRINT
2560  FOR J=1 TO 10
2570  ZS$(J)=SS＃(J)−S＃(J)
2580  ZD＃(J)=ZS＃(J)＊ST(J)
2600  LPRINT USING "＃＃"; J ;
2610  LPRINT N$(J) ;
2620  LPRINT USING "＃＃＃＃  ＃＃＃＃       ＃＃＃＃＃＃＃＃＃，"; ST(J), ZS＃(J), ZD＃(J)
2630  NEXT J
2640  ZZ＃=SZ＃−Z＃
2650  LPRINT "______________________________"
2660  LPRINT SPC(17) ;
2670  LPRINT USING "¥¥＃＃＃＃＃＃＃＃＃，"; ZZ＃
2680  INPUT "スミマシタラ キーイン", B$
2690  END
```

〔演習〕

売上一覧表（図3）の出力形式をコード，商品名，数量単価，金額と売上総金額となるよう，プログラムを改良してください．



プログラム7-3

```
10  WIDTH 80, 25 : CONSOLE 0, 25, 0, 1
20  PRINT CHR$(12)
30  LINE(0, 0) −(159, 99), PSET, 1, B
40  LINE(20, 10) −(50, 30), PSET, 4, B
50  LINE(80, 50) −(140, 80), PSET, 6, BF
60  FOR I=1 TO 1000 : NEXT I
70  LINE(20, 10) −(50, 30), PRESET, , B
80  LINE(80, 50) −(140, 80), PRESET, , BF
90  GOTO 40
```

3 論理演算

AND, OR, NOTなどの論理演算子も前述の関係演算子＞，＜，＝などと同様，２つの変数や定数，式の間で使われます．

❶数値変数Xの範囲を制限したい場合

たとえば，Xが０から９までの範囲の値をとるように制限をするには，次のようにする(AND)．

〔例〕

プログラム7-4

```
10  INPUT X
20  IF 0=<X AND X<=9 THEN 30 ELSE 10
30  PRINT X
40  END
```

すなわちANDは『かつ』という意味で，ここでは，Xが０より大きいか等しい，かつ，Xが９に等しいかそれより小さいのどちらも同時に成り立つならば，すなわち０≦X≦９の範囲内ならば30のステートメントへ飛び，そうでなければ10のステートメントへ飛ぶことを意味しています．

〔例〕　プログラム7-5

```
10  INPUT X
20  IF X<0 OR 9<X THEN 10 ELSE 30
30  PRINT X
40  END
```

すなわち，ORは『または』という意味で，Xがゼロより小さいか，または，９より大きいならば，すなわち，どちらか一方でも成り立てば，行番号10の文へ飛び，どちらも成立しなければ行番号30の文へ飛びます．

❷～でない場合，NOTの使用

〔例〕　プログラム7-6

```
10  INPUT X
20  IF NOT X=0 GOTO 10
30  PRINT X
40  END
```

❸ビット演算

〔例〕

AND操作

?15 and 12

12……………結果

ここで15＝1111（２進），12＝1100(２進）で表わされ，２つの数値の共に１のビットがその答となる．

〔例〕

OR操作

?5 or 3

7……………結果

ここで５＝0101(２進)，３＝0011(２進)で表わされ，２つの数値のすべての１のビットがその答となる．

〔例〕

NOT操作

?not 5

−６…………結果

５＝0101（２進）であり，NOT X＝−(X＋1)となって，２の補数表示となる．すなわち−６となる．

まとめ

以上の論理演算を２変数A，Bについてまとめると次のようになります．

〔注〕

〝真″または〝偽″の値を計算する論理演算子や関係演算子を含んだ式を**論理式**といいます．論理式の値は０のとき〝偽″，０でないとき〝真″と判定されます．

図6　各種の論理演算

変数		論理演算。					
A	B	AND	OR	NOT A	XOR	IMP	EQU
0	0	0	0	1	0	1	1
0	1	0	1	1	1	1	0
1	0	0	1	0	1	0	0
1	1	1	1	0	0	1	1

まとめ　演算の優先順位

① (　　)でくくられた式
② ∧ (ベキ乗)
③ − (負数)
④ ＊, ／
⑤ ¥ (整数の除算)
⑥ MOD (整数の剰余)
⑦ +, −
(①〜⑦ 算術演算子)
⑧ =, >, <など　関係演算子
⑨ NOT
⑩ AND
⑪ OR
⑫ XOR
⑬ IMP
⑭ EQU
(⑨〜⑭ 論理演算子)

同順位のものは左から右へ向って実行される.

4 数値関数の計算

初等関数である三角関数や対数関数はすでにBASICの中に組み込まれており，これらを組み込み関数といっています．sin cos tanなどはすでに，第2日目のプログラム2-5でもお目にかけましたが，そのほか絶対値をとるものや，整数化するものがあるので，これらについて実験してみましょう．

〔例〕

絶対値化

?abs(3−8) RET

5……………結果

〔例〕

整数化

?int(2.25), int(−4.32) RET

2　　　　−5…………結果

〔例〕

引数の整数部分のみ整数化

?fix(3.14),fix(−4.32) RET

3　　　　　−4……………結果

〔例〕

平方根

?sqr(2), sqr(3) RET

1.41421　1.73205……………結果

〔例〕

指数関数（*e*のベキ乗）

?exp(3.4) RET

29.9641……………結果　$e^{3.4}$の意味

〔例〕

自然対数

?log(3) RET

1.09861……………結果

〔例〕

符号のチェック

?sgn(5) : ?sgn(0) : ?sgn(−3)

1
0　……………結果
−1

〔例〕

三角関数

引数はラジアン単位なので度で表わすときは，ラジアンの値＝度×π／180に換算したものを引数にする．30°の場合，

プログラム7-7

```
10  K=30 *3.14159/180
20  PRINT "SIN="; SIN(K)
30  PRINT "COS="; COS(K)
40  PRINT "TAN="; TAN(K)
50  END
```

結果は次の通りとなる．演算は単精度である．

```
SIN=.5
COS=.866026
TAN=.57735
```

〔例〕

数値の形式変換

小数点以下切り捨ての整数化……………………………CINT(X)
単精度形式の数値化………………………………… CSNG(X)
倍精度形式の数値化…………………………………CDBL(X)

(1) **? cint(84.36)**

84……………結果

(2) **x＃=496.12348＃ : ? csng(x＃)**

496.123……………結果

(3) **x=432.1 : ? cdbl(x)**

432.1000061035156……………結果

〔演習〕

INT, FIX, CINTの各関数の相違点についてまとめる．

〔演習〕

SEC, CSC, COT（セカント，コセカント，コタンジェント）などの関数を作れ．

〔演習〕

0と1の間の乱数をうるRND関数を用いて0〜100までの乱数を10個作れ．



素数を求めるプログラムです．

100までの数のうちどのような素数があるでしょう．

プログラム7-8

```
10  REM ＊＊＊ PRIME NUMBER ＊＊＊
20  REM ＊＊＊ Eratosthenes' sieve ＊＊＊
25  INPUT "カズ ノ ハンイ U="; U : V=INT(U/2)
26  PRINT "U="; U; " "; "V="; V
30  DIM N(U), P(V)
40  FOR I=0 TO U
50  N(I)=0
60  NEXT I
70  K=0
80  FOR P=2 TO U
90  IF N(P)<0 THEN 150
100  K=K+1 : P(K)=P
110  IF P>SQR(U) THEN 150
120  FOR I=P TO U STEP P
130  N(I)=-1
140  NEXT I
150  NEXT P
155  PRINT
157  PRINT "＊＊＊＊＊"; U; "マデ ノ ソスウ ヒョウ ＊＊＊＊＊" : PRINT
160  FOR I=1 TO K
170  PRINT USING "＃＃＃＃"; P(I);
180  NEXT I
```

図7　出力結果

```
*****  100 マデ ノ ソスウ ヒョウ *****
2  3  5  7  11  13  17  19  23  29  31  37  41  43  47  53  59  61  67  71
73  79  83  89  97
```

8 第8日 ユーザー関数と乱数

〔目的〕

同じ式をプログラムの中で何度も使用するような場合には，DEF FNを使ってユーザー関数を定義します．sinやcosなどの三角関数は組み込み関数といって，すでにBASICインタープリタの中に準備されていました．これら組み込み関数以外のものはユーザーが作る関数ということになります．

関数を使うことの利点は引数を与えるだけで，その式の値が得られることであり，また，引数のリストの中に式も書くことができることです．

つぎに，乱数の発生についてはRND関数を用います．引数の値が正，負，ゼロにより，発生する乱数の系列が違ってくるので，これらについて，実験をします．ここでは，ユーザー関数の定義のしかたと乱数の発生プログラムについて学習します．

1 ユーザー関数

❶数値関数の作り方

DEF FN <名前>（<引数>）=<式>
（FN <名前>：ユーザー関数名）

引数は変数名です．複数個のときはカンマで区切ります．定義された関数を呼び出して使う方法は組み込み関数のときと同じです．

〔例〕

プログラム8-1

```
10  DEF FNA(X)=A*X*X+B*X+C
20  INPUT A, B, C
30  FOR X=1 TO 10
40  Y=FNA(X)
50  PRINT X ; Y
60  NEXT X
70  END
```

このプログラムは$Y=ax^2+bx+c$という関数を行番号10で定義し，係数a，b，cを入力し，Xを1から10まで増分1で変化させながらyの値を求めたものです．定義した関数の呼び出しは行番号40で行なっています．

〔例〕

プログラム8-2

```
100  DEF FNB(X, Y)=X^3+Y^2
110  INPUT A, B
120  K=FNB(A,B)
130  L=FNB(2, 3)
140  PRINT K, L
150  END
```

このプログラムでA＝4，B＝2を入力すると，行番号120の関数FNBの実引数は4，2となり，仮引数X，Yと置き替えられます．また，行番号130の関数呼び出しでは実引数として定数2，3が使われています．KとLの値はそれぞれこの場合には，68と17となります．

ここで注意しなければならないことは，関数定義文，

DEF FN <名前>（<引数>）=<式>

は，関数呼び出し　文より以前に定義しておくことが必要です．

❷文字関数の作り方

ユーザー関数名として文字型も許されるので文字関数も作ることができます．すなわち，

DEF FN <名前> $

となります．

〔例〕

プログラム8-3

```
200  DEF FNP$(X$)=CHR$(7)+X$+CHR$(13)
210  FOR I=1 TO 3
220  READ X$
230  PRINT FNP$(X$)
240  FOR J=1 TO 500 : NEXT J
250  NEXT I
260  DATA "TOKYO", "NAGOYA", "OSAKA"
270  END
```

このプログラムを実行すると画面にTOKYO，NAGOYA，OSAKAと順々に改行して出力します．出力の都度，内蔵のブザーを鳴らします．この関数呼び出しでは，ブザーを鳴らすこと，引数X＄をプリントし，改行することを行なっています．

〔例〕

プログラム8-4

```
300  DEF FNT$(X$, N$)=STRING$(5, 42)+"␣"+X$
     +"␣␣"+N$+"␣"+STRING$(5, 42)+CHR$(13)
     +DATE$+"␣␣"+TIME$
310  INPUT X$, N$
320  PRINT FNT$(X$, N$)
330  END
```

このプログラムはプログラム作成時にプログラム・リストの先頭にプログラム名と作者名，年月日，時刻を出力する文字関数です．X＄にプログラム名をN＄に作者名を入力します．

2 乱数の作り方

ゲームでのサイコロの目，トランプなどの切り札，その他，スーパーのレジ台の行列などマイコンでこれらの現象をシミュレーションするときによく使われる乱数について

学習しましょう．

〔例〕　　**プログラム 8 - 5**

```
100  FOR I=1 TO 5
110  PRINT RND(1)
120  NEXT I
140  PRINT "-------------"
150  FOR J=1 TO 5
160  PRINT RND(0)
170  NEXT J
180  PRINT "*************"
190  FOR K=1 TO 5
200  PRINT RND(-1)
210  NEXT K
1000 END
```

このプログラムは，RND(1)，RND(0)，RND(−1) を確かめてみるものです．実行結果は，図１の通りです．いずれも０と１の間の乱数を与えます．

引数が負のときは，新しい乱数の系列を作り，正のときは同じ系列の次の乱数を与え，ゼロのときは１つ前に発生した乱数を与えています．通常は乱数として正の数，RND(1)を使います．

〔演習〕

500個の乱数を発生するようにプログラムを改良し，実験してみてください．

図１　RND関数

```
.591065
.207991
.550967        RND(1)
.635863
.10411
-------------------------
.10411
.10411
.10411         RND(0)
.10411
.10411
****************
.522223
.522223
.522223        RND(-1)
.522223
.522223
```

〔例〕

金貨を投げて裏か表の出現する現象をマイコンでシミュレーションするプログラムを作ってみると，次のようになります．

プログラム 8 - 6

```
400  IF RND(1)<1/2 THEN 410 ELSE 420
410  COLOR 4 : PRINT "オモテ ダ！！" : GOTO 430
420  COLOR 2 : PRINT "ウラ デスヨ"
430  IF INKEY$=" " THEN 400 ELSE 430
1000 END
```

行番品400では，RND(1)の値が0.5以下かどうかを判定して，います．行番号430のINKEY$ 関数はキー入力としてこの場合，スペース・キーを押すと行番号400へ分岐します．キー入力がないときは，ヌルを入れてつぎの文に進みますので，この場合は，キー入力をしない限り，行番号 430へ分岐し，プログラムをこの位置で停止させているような状態にしています．

〔例〕　　**プログラム 8 - 7**

```
240  FOR I=1 TO 50
250  X=INT(RND(1)*100)
260  PRINT USING "####" ; X ;
270  NEXT I
1000 END
```

このプログラムは10進数２桁の乱数を50個発生し，印刷するものです．行番号250の文は，RND(1)*100でまず0〜99.9999までの乱数を作り，しかる後，INT 関数で整数化しています．

出力の結果例を図２に示しておきます．

図２　２桁の乱数

```
59 20 55 63 10 80  4 95 55 44  1 54 53 31 46 10 97 14
28 23 11 45 55 80 12 47 86 18 61 57 62 74 61 39 32  0
51 39 35 67 66  4 33 74 19 39 42 83 48
```

〔演習〕

サイコロをふる現象をシミュレーションしてみましょう．サイコロの目は１から６のいずれかなので，INT(RND(1)*6+1)とします．ここで１を足しているのは，０を出現させないためです．プログラムはつぎのようになります．

プログラム 8 - 8

```
300  FOR I=1 TO 200
310  X=INT(RND(1)*6+1)
320  COLOR X
330  PRINT X ;
340  NEXT I
1000 END
```

行番号320のカラー文を入れると，サイの目とカラーの色が一致して表示されます．

〔演習〕

４桁の乱数を100個発生できるプログラムを作ってください．

コーヒータイム 11

数当てゲームのプログラム

プログラム 8 - 9

```
100  X=INT(RND(1)*100)
110  INPUT "サテ イクツデショウ" ; N
120  1F N>X THEN PRINT "オオキイヨ" : GOTO 110
130  1F N<X THEN PRINT "チイサイヨ" : GOTO 110
140  1F N=X THEN PRINT "アタリ" : PRINT
150  INPUT "モウイチド ヤリマスカ(Y)/(N)" ; A$
160  1F A$="Y" THEN 100
170  PRINT "ゲーム オワリ"
180  END
```

9 第9日 文字列の取り扱い

〔目標〕

計算機は数値を使っていろいろな計算をし，実用に供されています．これらの分野は数値処理分野ですが，数値ばかりでなく，文字やパターンなど非数値データを処理することもできます．

マイコンで実用化が急速に進んでいる電子翻訳機や，音声出力機器での処理は，文字処理が中心になります．そのほか，最近，にぎやかになったＯＡ機器のうちワード・プロセッサ（文章作成機）はまさしく文字処理が中心的技術になります．

ここでは，文字列の交換，挿入，切り出し，置換，などを行なえるいろいろなストリング関数について学習します．最後に文章をコード化して発信し，数値で受信したものを再び文章化するプログラムについて少しふれます．

1 文字を文字コード(ASCIIコード)にする

計算機の内部では文字は定められた文字コードで格納されます．たとえばキーボードから〝A〟を入力すると〝65〟に変換されて記憶されます．

〔演習〕ＡＳＣ関数

プログラム１を実行していろいろなキー入力を行ない．出力結果を確かめてください．**図１**は出力結果例を示します．

プログラム９-１（ＡＳＣ）

```
10 INPUT "A-Z";A$
20 PRINT A$,ASC(A$)
30 GOTO 10
```

図１　出力結果

```
A    65
B    66
C    67
D    68
E    69
F    70
G    71
H    72
I    73
J    74
K    75
L    76
!    33
#    35
$    36
%    37
&    38
,    39
(    40
)    41
```

2 文字コードを文字にするには

計算機内部での文字コード〝65〟を文字に変換するには〝CHR$(65)〟とします．すると文字〝A〟となります．

〔演習〕ＣＨＲ関数

プログラム２を実行してください．ここで行番号"CHR$(34)"は引用符（〝）です．行番号30の**RIGHT$**，**STR$**は後述します．出力結果例を**図２**に示します．**CHR$(49)** は数字〝１〟，**CHR$(42)** は〝＊〟などとなります．

プログラム９-２（ＣＨＲ＄）

```
10 C$=CHR$(34)
20 FOR I=32 TO 63
30 I$=RIGHT$("  "+STR$(I),3)
40 LPRINT "CHR$("I$")="C$CHR$(I)C$,
50 NEXT I
```

3 文字型データと数値型データの変換

文字型データＸ＄と数値型データＸはＸ＄＋Ｘなどの式としては許されず，エラーになります．文字型データを数値型データに変換するには**VAL(X$)**とします．

逆に，数値型データを文字型データに変換するには，**STR$(X)**とします．

〔演習〕ＶＡＬ関数，ＳＴＲ関数

プログラム９-３を実行してください．Ａ＝＋12345，E$＝"12345"で前者は数値です．後者は文字列としての12345です．**図３**に出力結果列を示します．

プログラム９-３

```
10 E$="12345"
20 X=67890
30 A=VAL(E$)
40 B$=STR$(X)
50 LPRINT A,B$
60 LPRINT E$,X
70 END
```

図３　出力結果

```
 12345                67890
12345                 67890
```

4 文字の切り出し

文字列の中から任意の文字を取り出したり，その文字列を前後に分解したり，先頭からいくつかの文字を取り出したり，文字列の最後から任意個の文字を取り出したりでき

図２　プログラム９-２の出力結果例

```
CHR$( 32)=" "   CHR$( 33)="!"   CHR$( 34)="""   CHR$( 35)="#"   CHR$( 36)="$"
CHR$( 37)="%"   CHR$( 38)="&"   CHR$( 39)="'"
CHR$( 40)=")"   CHR$( 41)=")"   CHR$( 42)="*"   CHR$( 43)="+"   CHR$( 44)=","
CHR$( 45)="-"   CHR$( 46)=","   CHR$( 47)="/"
CHR$( 48)="0"   CHR$( 49)="1"   CHR$( 50)="2"   CHR$( 51)="3"   CHR$( 52)="4"
CHR$( 53)="5"   CHR$( 54)="6"   CHR$( 55)="7"
CHR$( 56)="8"   CHR$( 57)="9"   CHR$( 58)=":"   CHR$( 59)=";"   CHR$( 60)="<"
CHR$( 61)="="   CHR$( 62)=">"   CHR$( 63)="?"
```

る関数があります．

〔演習〕RIGHT関数，LEFT関数

プログラム9-4 (RIGHT，LEFT)

```
10 S$="123456789"
20 FOR I=0 TO 15
30 PRINT RIGHT$(S$, I), LEFT$(S$, I)
40 NEXT I
50 END
```

図4 出力結果

```
9                1
89               12
789              123
6789             1234
56789            12345
456789           123456
3456789          1234567
23456799         12345678
123456789        123456789
123456789        123456789
123456789        123456789
123456789        123456789
123456789        123456789
123456789        123456789
123456789        123456789
```

プログラム9-4を実行してください．

RIGHT$(S$,I)は文字列S$（=123456789）の右端からI個ずつの文字を切り出します．

LEFT$(S$,I)は文字列S$の左端からI個ずつの文字を切り出します．出力結果例を図4に示します．

〔演習〕MID関数

プログラム9-5 (MID$ LEN)

```
10 S$="123456789"
20 FOR I=0 TO 15
30 X$=MID$(S$, 3, I)
40 PRINT X$, LEN(X$)
50 NEXT I
60 END
```

図5 出力結果例

```
              0
3             1
34            2
345           3
3456          4
34567         5
345678        6
3456789       7
3456789       7
3456789       7
3456789       7
3456789       7
3456789       7
3456789       7
3456789       7
3456789       7
```

プログラム9-5を実行すると，図5に示すような結果が得られます．MID$(S$, 3, I)の意味は文字列S$の左端から3番目の文字からI個取り出すことを意味します．Iが0から15まで変化していきますが，文字列の並びがIより小さいときは右端まですべて切り出されます．

なお，MIDにはMID$(S$,x,y)=A$なる型のものがあり，文字列の一部を置き換えることもできます．すなわちS$の並びのうち$x$番目から$y$文字をA$の並びで置き替えます．

プログラム9-6

```
100 X$="123456789"
110 MID$(X$,3,4)="BOOK"
120 Y$="123456789":A$="I/O"
130 MID$(Y$,5)=A$
140 PRINTX$:PRINTY$:END
```

図6 出力結果

```
12BOOK789
1234I/O89
```

行番号130のようにMID関数の第3番目の引き数を省略する場合もあります．

5 文字列の長さ

文字列の文字数を知りたいときがあります．そのときに使うのがLEN関数です．

〔演習〕LEN関数

プログラム9-5の行番号40によってLEN(X$)が表示されます．図5の出力結果例の右側がLEN(X$)の値です．

6 同じ文字をいくつか並べたいとき

タイトルなどを表示するときに，“*”印や“−”印をいくつか使用して枠を作ったりします．また，特に，プログラム・リストを見やすくするためにもよく使います．そのようなときにSTRING関数を使うこともできます．

〔演習〕STRING関数

プログラム9-7，9-8を実行すると，それぞれ図7，図8のような結果が得られます．

ここで，STRING$(10, 45)とは文字コード“45”の文字を10個出力するということです．

プログラム9-7

```
10 T$=STRING$(10,45)
20 PRINT T$;" STRING$(10,45 ) TEST ";T$
30 END
```

図7 出力結果

```
----------STRING$(10,45) TEST ----------
```

プログラム9-8

```
10 T$=STRING$(10,42 )
20 PRINT T$;" STRING$(10,42 ) TEST ";T$
30 END
```

図8 出力結果

```
**********STRING$(10,42) TEST **********
```

〔演習〕

プログラム9-9はSTRING関数の引数の値をキー入力するようにしたものです．任意の文字，文字数を入力し，表示することができます．図9は#，%，*の出力結果を示しました．文字として文字コード“32”を与えると空白を意味し，次のSPACE関数と同じ結果になります．

プログラム9-9

```
10 INPUT "STRING$(N, C$) TEST N=,S$=";N,C$
20 T$=STRING$(N, C$)
30 PRINT T$;"STRING$("N;",";C$;") TEST";T$
40 END
```

図9　出力結果

```
##########STRING$(10,#)TEST##########
%%%%%%%%%%STRING$(10,%)TEST%%%%%%%%%%
**********STRING$(10,*)TEST**********
```

〔演習〕SPACE関数

プログラム9-10

```
10 FOR I=0 TO 2
20 A$=SPACE$(I)
30 PRINT A$; CHR$ (I+65)
40 NEXT I
```

図10　出力結果

```
A
 B
  C
```

7 文字列の探索

文字列A$の中に文字列B$が含まれているかどうか調べたいときは**INSTR(A$,B$)**関数があります．

この関数は文章の中で人名を探したり，商品一覧表から，特定の商品があるかないかを探したりするときなどに使われます．含まれていればその文字列中での位置を値とし，含まれていないときはゼロを値とします．

〔演習〕INSTR関数

プログラム9-11

```
100 A$="ABCDEFGABCD"
120 B$="H":I=INSTR(A$, B$)
130 C$="D":J=INSTR(A$, C$)
140 D$="F":K=INSTR(A$, D$)
150 PRINT I;J;K
160 END
```

プログラム9-11を実行すると，図11のような出力結果になります．

図11　出力結果

```
0 4 6
```

プログラム9-11ではDを見つけ出すのに行番号130で**INSTR**関数を使いました．しかし，4番目のDだけが探し出されましたが，最後の位置にあるDは探し出されませんでした．任意の位置から探索を始める関数が**INSTR(P,A$,C$)**です．

Pは文字列A$のP文字目から探索を始める位置を指定しています．

プログラム9-11の行番号130を次のように変更して実行結果を確認してください．

```
130 C$="D":J=INSTR(7, A$, C$)
```

表示結果は0　11　6となりましたが，Dの位置をすべて表示させるには次のようにしなければなりません．

〔演習〕

プログラム9-12

```
200 A$="ABCD ABCD ABCD"
210 I=0
220 I=INSTR(I+1, A$, "D")
230 PRINT I;
240 IF I< >0 THEN 220
250 END
```

このプログラムでは出力結果は図12のようになりA$の中の〝D〟の位置が複数個みつかります．

図12

```
4 8 12
```

8 日付と時刻の表示

81年09月30日と画面に表示させるには**DATE$**を，12時30分50秒と表示させるには**TIME$**を使って，日付や時刻を文字型データとして切り出します．

〔演習〕

プログラム9-13

```
300 INPUT "年/月/日";A$
310 INPUT "時/分/秒";B$
320 INPUT "ナマエ";N$
330 DATE$=A$:TIME$=B$
340 PRINT DATE$
350 PRINT TIME$
360 Y$=LEFT$(DATE$, 2)
370 M$=MID$(DATE$, 4, 2)
380 D$=RIGHT$(DATE$, 2)
390 H$=LEFT$(TIME$, 2)
400 F$=MID$(TIME$, 4, 2)
410 S$=RIGHT$(TIME$, 2)
420 PRINT
430 PRINT Y$;"年",M$;"月";D$;"日"
440 PRINT H$;"時";F$;"分";S$;"秒"
450 PRINT "BY␣";N$
460 END
```

プログラム9-13はプログラム開発時にプログラム・リストをとるたびに日付，時刻，作成者名を最初に印刷しておくためのものです．

プログラム・リストの整理のためにも便利なプログラムです．

まとめ

文字列を処理するためのいろいろな関数をまとめたものが表1です．

〔演習〕

アルファベット大文字[A]～[Z]と[スペース]を入力することができ，スペースは00，Aは01にBは02に，以下Zは26まで，それぞれ数値に変換するプログラムを作ってみましょう．

出力は，入力した文字列（発信文）とその長さです．変換した数値は配列P(X)へ格納します．

表1　文字列処理のための関数，ステートメント

機　　能	書　　式	内　　容
文字を入力する	INPUT A＄	255文字までの文字データを入力できる．CR ，コントロール・コード入力不可
	LINE INPUT A＄	キャリッジ・リターン・キーが押されるまでの１行全部を入力できる．
	A＄＝INKEY＄	キーが押されていればその文字を，押されていなければヌル・ストリングが与えられる．
	READ A＄	DATA文で与えられた文字データを入力する．
	A＄＝INPUT＄(N)	長さＮ文字の文字列を入力する．Ｎ文字入力するまでキーボード待ちとなっている．
文字列の長さ	LEN(A＄)	文字列の長さにはプリントされない符号やスペースなども数える．
文字のコード化	ASC(A＄)	文字列A＄の最初の文字のコードが与えられる．
文字コードの文字への変換	A＄＝CHR＄(N)	文字コードNに対応する文字を与える．
数値を数字列に変換	A＄＝STR＄(N)	数値Ｎを文字列Ａ＄に変換します．
数字列を数値に変換	N＝VAL(A＄)	文字の並びを数値に変換します．
文字列の左部分の取り出し	LEFT(A＄，N)	左端からＮ文字を取り出す．
文字列の右部分の取り出し	RIGHT(A＄，N)	右端からＮ文字を取り出す．
文字列の抜き出し	MID＄(A＄，N，X)	文字列Ａ＄のＮ番目の文字からＸ個の文字を抜き取る．
文字列の挿入，置き換え	MID＄(A＄，N，X)＝B＄	文字列Ａ＄のＮ番目の文字からＸ文字を文字列Ｂ＄で置き換える．または，文字列Ａ＄のＮ番目から文字列Ｂ＄を挿入する．
文字列の探索	INSTR(A＄，B＄)	文字列Ａ＄の中で文字列Ｂ＄と同じ文字列部分を探し，合致した最初の文字が，Ａ＄の何番目かを与える．
	INSTR(N，A＄，B＄)	文字列Ａ＄のＮ番目から探索を開始し，Ｂ＄と同じ文字列があるかどうか探す．結果はINSTR(A＄，B＄)と同じ．
同一文字の並びを作る	A＄＝STRING＄(X，N)	文字コードＮの文字をＸ個並べる．
スペースを複数個作る	SPC(X)	PRINT文中でＸ個の空白部分を作る．
	A＄＝SPACE＄(X)	Ｘ個の空白からなる文字列を与える．
指定位置まで空白を作る	TAB(X)	PRINT文中でカーソルをＸ桁まで移動する．
16進変換	A＄＝HEX＄(N)	Ｎの16進表現をする文字列を与える．
８進変換	A＄＝OCT＄(N)	Ｎの８進表現をする文字列を与える．

プログラム９-14

```
  10 DIM P(260)
1000 PRINT CHR$(12)
1010 INPUT "ハッシンブン ヲ ニュウリョク シテ クダサイ";M$
1020 L=LEN(M$):PRINT
1030 PRINT "/";M$;"/"; "";L; "モジデス"
1040 B=L MOD 2
1050 IF B <>0 THEN M$=M$+" ":L=L+1
1060 FOR X=1 TO L
1070 MI$=MID$(M$, X,1)
1080 IF MI$=" " THEN P(X)=ASC(MI$)-32:GOTO 1100
1090 P(X)=ASC(MI$)-64
1100 NEXT X
```

入力された文字列はＭ＄に格納され，**LEN(M$)** で長さが与えられます．

1040～1050では文字列の文字数が奇数個のときはスペースを１個追加しています．

1060～1100は**MID$(M$,X,1)**を使って，入力した文字列から１文字ずつ切り出し**ASC(MI$)**によって文字コードに変換され，０～26の数値に変換しているところです．

1090で変換された数値は配列Ｐ(Ｘ)に格納しています．

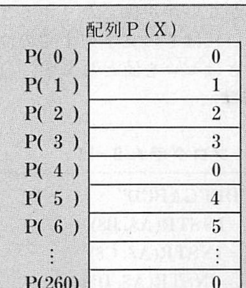
〔演習〕

配列Ｐ(Ｙ)に格納されている数値を引き続く２組を単位として，数字列に変換する．変換した結果を表示し，スペース・キーが押されるまで画面を保持しているプログラムを作ります．数値範囲は０～26とします．

図13　M＄＝ABC―DE を入力した場合の出力結果

```
/ABC DE/ 6モジデス
```

図14



プログラム９-15は1140で，**STR$(P(Y))**によってまず数値ひとつをＦ＄という数字列に変換しています．いま，ここでは数値は０～26の範囲とするので，変換された数字列は00～26のいずれかになります．**LEN(F$)**の値が２ということは数値が１桁ということで，**MID$**関数を使って，〝＋〟符号桁のところを〝０〟と置き換えています．

同様に**LEN(F$)**の値が〝３〟ということは数値２桁であることを意味するので，この場合は右から２文字分を取り出しています．数字列に変換されたら配列**M$(J)**に格納し，画面にその結果を表示しています．

プログラム9-16は画面出力結果を保持するためのものです．

配列Ｐ(Ｙ)，Ｍ＄(Ｊ)の対応例は図15に示す通りです．

〔演習〕

数値配列Ｍ(Ｉ)の値を文字列に変換し，受信文を作るプログラムを考える．数値は４桁以内とし，00はスペース，

プログラム 9-15

```
  20 DIM M$(150), P(260)
1110 Y=0:W=L/2:PRINT
1120 FOR J=1 TO W
1130 Y=1+(J-1)*2
1140 F$=STR$(P(Y)):IF LEN(F$)=2 THEN MID$(F$,1,1)="0"
1150 IF LEN(F$)=3 THEN F$=RIGHT$(F$,2)
1160 B$=STR$(P(Y+1)):IF LEN(B$)=2 THEN MID$(B$,1,1)="0"
1170 IF LEN(B$)=3 THEN B$=RIGHT$(B$,2)
1180 M$(J)=F$+B$
1190 PRINT M$(J);
1200 NEXT J:PRINT
1210 GOSUB 4500
1220 END
```

プログラム 9-16

```
4500 COLOR 5
4510 PRINT: PRINT: PRINT "ハジメニ モドルニハ スペースキー ヲ オシテクダサイ"
4520 IF INKEY$<>" " THEN GOTO 4520
4530 COLOR 7:RETURN
```

図15

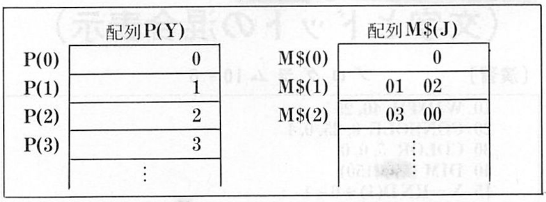


図16 A B C ─ D E に対する出力

```
0102   0300  0405  0102   0300  0405
```

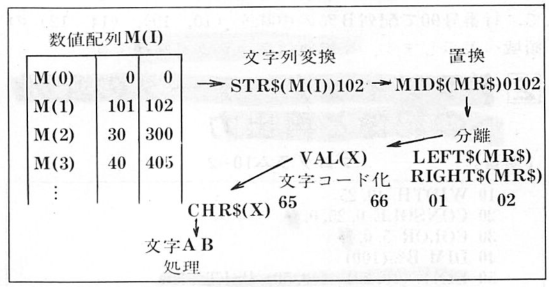
プログラム 9-17

```
  30 DIM M(150)
3000 REM ジュシンブン ノ サクセイ
3010 PRINT "ジュシンブン ハ ツギノ トウリデス" :PRINT
3020 FOR I=1 TO W
3030 MR$=STR$(M(I))
3040 LR=LEN(MR$)
3050 IF LR>=5 THEN K=3:GOTO 3090
3060 IF LR=3 THEN MID$(MR$,1,1)="0":K=1: GOTO 3090
3070 IF LR=4 THEN MID$(MR$,1,1)="0":K=2:GOTO 3090 IF
3080 IF LR=2 THEN MID$(MR$,1,1)="0":MR$="00"+MR$:K=2
3090 L$=LEFT$(MR$,K):R$=RIGHT$(MR$,2)
3100 F=VAL(L$)+64:B=VAL(R$)+64
3110 IF F=64 THEN F=32 ELSE IF B=64 THEN B=32
3120 COLOR 3
3130 PRINT CHR$(F);CHR$(B);
3140 NEXT I:COLOR 7
3150 END
```

1はAに，2はBに以下26はZに変換するものとする．配列M(I)の1つには2文字分が格納されているものとする．

プログラムは**プログラム 9-17**のとおりです．ここでは，

図17 プログラム17の内容



3030で数値配列の要素を文字に変換しています．

文字の長さをLEN(MR$)で調べ符号を考慮して3059～3080で4文字になるように置換しています．

3090でそれぞれ右2文字，左文字を分離し，文字列を数値に3100で変換しています．

3110はスペースを検出するものです．

3130ではASCIIコードに変換されたFやBが文字になって表示されます．図17にはこの様子が図式で表わされています．

コーヒータイム 12

文字列M$の中から文字列X$を探して，文字列Y$と入れ替えるプログラム．

プログラム18

```
100 M$="ミカン␣メロン␣スイカ␣トマト␣"
110 PRINT M$
120 INPUT "X$=";X$
130 INPUT "Y$=";Y$
140 I=1
150 I=INSTR(I,M$,A$)
160 IF I THEN MID$(M$,I)=Y$:GOTO 150
170 PRINT:PRINT M$
180 END
```

10 第10日 グラフィックス入門

〔目標〕

マイコンで絵を描くにはどうしたらよいのでしょう．幸い，PSETとか，LINE，GET，PUTなどというグラフィック命令が準備されており，これらの命令を使用すると，簡単なグラフィックが容易に実現できます．

ここでは，これらのグラフィック命令の機能について学習します．実際の複雑な絵はこれらの命令を組み合わせて創作します．各命令の機能が確認できたら，一度，各自が実際に絵を作ってみることをお推めします．

GET，PUTはPC-8001独自のグラフィック命令です．

1 文字データの配列への記憶と再出力

〔演習〕 プログラム10-1

```
 10 WIDTH 40,25
 20 CONSOLE 0,25,0,0
 30 COLOR 0,0,0
 40 DIM B%(40)
 50 LOCATE 0,0:PRINT "■■■■■"
 60 LOCATE 0,1:PRINT "■■■■■"
 70 LOCATE 0,2:PRINT "■■■■■"
 80 GET @(0,0)-(4,2),B%
 90 PUT @(10,10)-(14,12),B%
100 END
```

このプログラムはGET，PUTを使って配列B%に"■"を格納し，再び出力するプログラムです．すなわち，画面上上（0，0）と（4，2）の座標で囲まれた領域に"■"を表示し，（50～70の行番号），行番号80で，その領域の画

面表示パターンをそのまま，配列Ｂ％に格納します．そして，行番号90で配列Ｂ％の中味を（10, 10)，（14, 12）の領域へ表示します．座標値はキャラクタ座標です．

②グラフィック・データの配列への記憶と再出力

〔演習〕　　　　　プログラム10-2

```
 10 WIDTH 40,25
 20 CONSOLE 0,25,0,1
 30 COLOR 5,0,1
 40 DIM B%(100)
 50 LINE (10,20)-(40,50),PSET,,BF
 60 GET @(10,20)-(40,50),B%,G
 70 FOR I=0 TO 1000:NEXT
 80 COLOR 4,0,1
 90 PUT @(30,40)-(60,70),B%, OR
100 END
```

行番号50で画面に四角形を描きます．行番号60のＧＥＴ文はドット座標値で配列Ｂ％に画面領域を格納します．行番号90では，ドット座標値の領域（30, 40)，(60, 70）に演算ORで出力します．

〔演算〕　　　　　プログラム10-3

```
 10 WIDTH 40,25
 20 CONSOLE 0,25,0,1
 30 COLOR 5,0,1
 40 DIM B%(100)
 50 LINE (10,20)-(40,50),PSET,,BF
 60 GET @(10,20)-(40,50),B%,G
 70 FOR I=0 TO 1000:NEXT
 80 COLOR 4,0,1
 90 PUT @(30,40)-(60,70),B%,AND
100 END
```

このプログラムも演算して出力するプログラムです．行番号90ではANDをとっています．

〔演算〕　　　　　プログラム10-4

```
 10 WIDTH 40,25
 20 CONSOLE 0,25,0,1
 30 COLOR 5,0,1
 40 DIM B%(100)
 50 LINE(10,20)-(40,50),PSET,,BF
 60 GET @(10,20)-(40,50),B%,G
 70 FOR I=0 TO 1000:NEXT
 80 COLOR 4,0,1
 90 PUT @(30,40)-(60,70),B%,XOR
100 END
```

このプログラムは XOR 演算を画面領域上で行ない出力します．

LINE文中のPSETはドットを表示することを示します．

PSETはそのほかに単独で使われます．

　　　　PSET(X, Y, C)

ここで座標（X, Y）のところにカラーコード〝C″でドットを表示します．〝C″は白黒モードではCOLOR文のときと同様な機能となります．

画面のドット表示を消すにはPRESET(X, Y)を使います．

③動く図形

行番号40でドット指定で箱を書きます．行番号50で配列Ｂ％に箱を格納します．行番号55では画面の箱を消します．行番号60で配列Ｂ％から箱を出してきて，違った領域へ出力します．あたかも箱が移動したかのように表示されます．

〔演習〕　　　　　プログラム10-5

```
10 WIDTH 40,25
20 CONSOLE 0,25,0,1
30 COLOR 5,0,1
35 DIM B%(50)
40 LINE (5,10)-(10,22),PSET,,BF
50 GET @(5,10)-(10,22),B%,G
52 FOR I=0 TO 1000:NEXT
55 PUT @(5,10)-(10,22),B%,PRESET
60 PUT @(45,50)-(50,62),B%,PSET
70 END
```

④色の変わる図形（文字とドットの混合表示）

〔演習〕　　　　　プログラム10-6

```
 10 WIDTH 40,25
 20 CONSOLE 0,25,0,1
 30 COLOR 5,0,0
 40 DIM B%(150)
 45 X=RND(1)*6+1
 50 LINE (10,20)-(40,50),PSET,X,BF
 60 GET @A(10,20)-(40,50),B%
 70 FOR I=0 TO 1000:NEXT
 90 PUT @A(30,40)-(60,70),B%
100 END
```

このプログラムはLINE文の機能コードに乱数を使い，いろいろな色で箱を書くようにしたものです．

GET@A，PUT@Aは文字とドット混合表示のときに使います．



プログラム10-7

```
 10 WIDTH 40,25
 20 CONSOLE 0,25,0,1
 25 DIM B%(150),A%(10)
 30 COLOR RND(1)*6+1,0,1
 50 LINE (10,20)-(40,50),PSET,BF
 60 GET @ (10,20)-(40,50),B%,G
 90 PUT @ (30,40)-(60,70),B%,XOR
100 LINE (0,0)-(5,5),PSET,BF
110 GET @(0,0)-(5,5),A%,G
120 PUT @(0,0)-(5,5),A%,PRESET
125 X1=0:Y1=0
130 X1=X1+1:Y1=Y1+1
135 COLOR RND(1)*6+1,0,1
150 PUT @(X1,Y1)-(X1+5,Y1+5),A%,PSET
160 FOR I=1 TO 300:NEXT
165 PUT @(X1,Y1)-(X1+5,Y1+5),A%,PRESET
170 IF X1<76 GOTO 130
180 GOTO 30
```

どんな絵が出てくるでしょうか，楽しみにプログラムを実行してみてください．

11 第11日　プログラム制御の方法

〔目標〕

第１日から第10日までいろいろな命令やプログラムを学習してきました．実用的なプログラムを作るには，これだけの知識ではまだ大変だと思います．プログラムは機能別

にできるだけ小さく作ってこれらの小さないくつかのプログラムを上手に結合し，ひとつの大きなプログラムにしていろいろな機能を実現するプログラムにまとめ上げて作ります．

ここでは，小さないくつかのプログラムをまとめてひとつの大きなプログラムとし，まとまったプログラムに作り上げるために必要な命令をいくつか学習します．

具体例としてはいままでに使ったいくつかのプログラムを結合した売上・在庫管理プログラムの模型と，暗号処理プログラムを作ることにします．

プログラム制御の要点はサブルーチンの作り方とプログラムの流れの作り方です．

1 プログラムの流れを変える

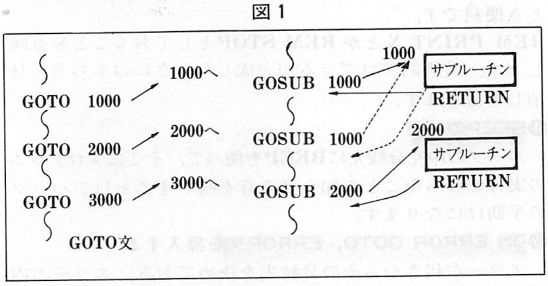
プログラムの流れを変えるには無条件分岐するGOTO文と条件によって分岐するON〈条件〉GOTO文があります．

そのほか，サブルーチンを使用したときのGOSUB文とON〈条件〉GOSUB文があります．これら２種類の文はサブルーチンへ分岐するか否かの違いです．

サブルーチンは，プログラムの中で繰り返し何度も使われる，まとまった小さなプログラムだと思ってください．そして，必らず，分岐していってもサブルーチン・プログラムでの処理が終り次第，分岐した直後の命令のところに実行の順番が自動的に戻ってくるものです．GOTO文は戻ってこないので，ここらあたりが違います．

このようすを図１に示します．サブルーチンから戻るにはRETURN命令が必要になります．

図１



〔演習〕ON X GOSUB

プログラム11-1

```
100 INPUT "X=";X
110 ON X GOSUB 200,300,400
120 GOTO 100
130 REM *** サブ ルーチン ***
200 PRINT "SUB-1"
210 RETURN
300 PRINT "SUB-2"
310 RETURN
400 PRINT "SUB-3"
410 RETURN
500 END
```

このプログラムを実行してみるとX＝１のとき200へ分岐し，SUB－１とプリントしてきます．X＝２のとき300へ分岐しSUB－２とプリントしてきます．

X＝３のとき400へ分岐しSUB－３とプリントしてきます．

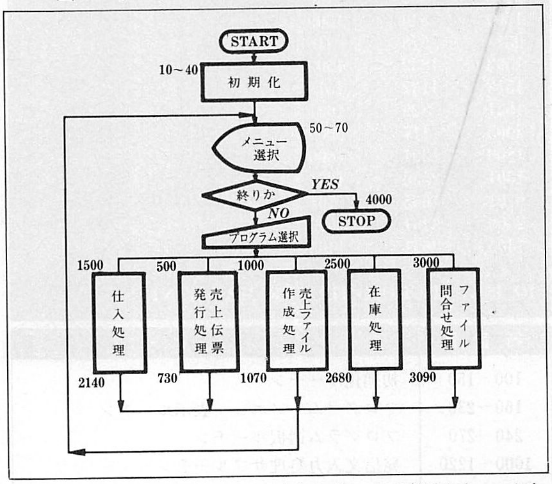
それぞれ戻るためにはRETURN命令が必要です．戻ってきたときは120のGOTO100へ戻ります．

2 プログラムの構成法

〔演習〕仕入，売上，在庫処理プログラム

仕入，売上，在庫の処理プログラムをまとめてひとつのプログラムにしてみよう．プログラムを機能別に箱に書き，それを結合する．そして，各機能プログラムをどのようにして選択して実行させるかを考える．

図２　仕入，売上，在庫処理プログラムの構成



プログラムの構成は図２に示すように，６個のブロックとプログラムになります．これら機能ブロックはそれぞれサブルーチン化し，行番号を適当に割り付けてコーディングします．

このようにしていままで学習してきたプログラムをまとめてみると行番号は図２に示したものとなります．

すなわち，

行番号	
10～40	初期化ルーチン
50～70	メニュー選択ルーチン
500～730	売上伝票作成サブルーチン
1000～1070	売上ファイル作成サブルーチン
1500～2140	仕入処理サブルーチン
2500～2680	在庫ファイル作成サブルーチン
3000～3090	ファイル問い合せ処理ルーチン

となります．

プログラム・リストを図３に示します．

〔演習〕暗号処理プログラム

図４　暗号処理プログラムの構成

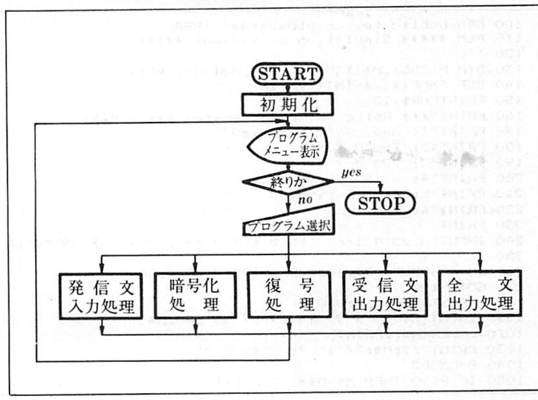


図6　出力結果例

```
*** ゼンブン ノ シュツリョク ***
      ハッシンブンハ
/IO IS A MICROCOMPUTER MAGAZINE/ 30 モジ デス
コード  アンゴウ  フクゴウ
 0915     624      915
 0009    2072        9
 1900    2342     1900
 0100    1952      100
 1309     840     1309
 0318     578      318
 1503    1832     1503
 1513    1132     1513
 1621     829     1621
 2005     117     2005
 1800    1391     1800
 1301    2556     1301
 0701    1665      701
 2609    2461     2609
 1405    2641     1405
ジュシンブン ハ ツギノ トオリデス
IO IS A MICROCOMPUTER MAGAZINE
```

行番号	
100～150	初期化ルーチン
160～230	プログラム・メニュー表示ルーチン
240～270	プログラム選択ルーチン
1000～1220	発信文入力処理サブルーチン
2000～2190	暗号化処理サブルーチン
2500～2620	復号処理サブルーチン
3000～3160	受信文出力処理サブルーチン
4000～4140	全文出力処理サブルーチン
4500～4530	画面表示保持サブルーチン
5000～	END

なお，出力結果リストの例は図6に示します．

文字処理システムの応用例として暗号処理プログラムを考えてみます．ここでは，1976年にヘルマンが発表した公開鍵暗号系についてRSA法によるアルゴリズムをプログラム化した例を図4にプログラムの構成を，図5にプログラム・リストを示しました．

このプログラムは発信文としてはスペース，A～Zの文字のみからなる文章が許されます．

暗号鍵e，nは公開でnは素数pとqの積で，eは乱数から作られます．

秘密にしておく復号鍵dは $e \cdot d \equiv 1 \bmod 〔(p-1)\cdot(q-1)〕$ から選ばれます．

このプログラムを実行するために $e=17$，$d=157$，$n=2773$ が例として示されています．

3 プログラムのデバックについて

プログラムは一度に正しいものができるとは限りません．何度も何度も手直ししてできあがります．特に，BASICのプログラムはそのようなことを前提にシステムが作られており，プログラムの手直しが非常にやり易くなっています．

次に，2～3ご参考までにプログラム・デバックのやり方を紹介しておきます．

❶STOP文を使う

プログラムの途中に適時STOP文を挿入し途中結果を確認します．このとき，変数の値を調べるにはダイレクト・モードでPRINT文などを使います．

❷TRON/TROFFを使う

実行した行番号を表示させればプログラムがどのように走ったか（実行されたか）判りやすくなります．実行順に行番号を表示するにはTRONをまた，その表示を中止させるにはTROFF命令を使います．FOR文や，GOTO文などでループを作るプログラムのトレースには有効な方法です．

❸PRINT文を使う

変数の値をプログラムの実行を中止することなく，順次続けて監視したいときは適当な個所にPRINT文を入れて出力させます．デバックのために挿入します．

STOP文や，PRINT文が不要になったらすぐに消すことなくREM文に置き換えておくのもまた，必要になったとき便利です．

REM PRINT XとかREM STOPとしておくことを意味します．最終的プログラムが完成したときにはもちろん抹消しておきます．

❹BEEPの使用

ループの中や分岐先にBEEPを使って，そこにプログラムの実行が移ったことを知らせる音を鳴らすことはデバックの手助けになります．

❺ON ERROR GOTO，ERROR文を挿入する

エラーが起きたらその分岐先を決めておき，エラーの内容を調べるようにします．

❻最後に初心者の方に多くみられますが，プログラムを訂正した場合は必らず RETURN キーを押しておくことを忘れないようにしましょう．

図5

```
100 REM public kev crvptosvstem  (RSA)
110 REM ***** Simulation bv K.Kawai *****
120 '
130 DIM P(255),M$(130),C(130),M(130),F$(1)
140 DEF FNMF(X)=X-INT(X/N)*N
150 PRINTCHR$(12)
160 PRINT"*** Pblic key cryptosvstem ***":PRINT
170 PRINT"1: ハッシン ブン ノ ニュウリョク"
180 PRINT"2: アンゴウ ノ サクセイ        "
190 PRINT"3: フクゴウ ノ サギョウ        "
200 PRINT"4: ジュシン ブン ノ サクセイ   "
210 PRINT"5: ゼンブン シュツリョク       "
220 PRINT"6: オワリ                      "
230 PRINT
240 INPUT"バンゴウ 1-6 ノ ドレカ ヒトツ ヲ キー ニュウリョク シテ クダサイ";A
250 '
260 ON A GOSUB1000,2000,2500,3000,4000,5000
270 GOTO 150
1000 PRINTCHR$(12):N=0
1010 INPUT"ハッシンブン ヲ ニュウリョク シテ クダサイ";M$
1020 L=LEN(M$):PRINT
1030 PRINT"/";M$;"/";" ";L;"モジ デス"
1040 B=LMOD2
1050 IF B<>0 THEN M$=M$+" ":L=L+1
1060 FORX=1TOL
1070 MI$=MID$(M$,X,1)
1080 IFMI$=" "THENP(X)=ASC(MI$)-32:GOTO 1100
1090 P(X)=ASC(MI$)-64
1100 NEXTX
1110 Y=0:W=L/2:PRINT
1120 FORJ=1TOW
1130 Y=1+(J-1)*2
1140 F$=STR$(P(Y)):IF LEN(F$)=2 THEN MID$(F$,1,1)="0"
1150 IFLEN(F$)=3 THEN F$=RIGHT$(F$,2)
1160 B$=STR$(P(Y+1)):IF LEN(B$)=2 THEN MID$(B$,1,1)="0"
1170 IF LEN(B$)=3 THEN B$=RIGHT$(B$,2)
1180 M$(J)=F$+B$
1190 PRINTM$(J);
1200 NEXTJ:PRINT
1210 GOSUB4500
1220 RETURN
2000 REM *** Public key cryptosystem ***
2010 REM Encryption and Decryption program
2020 REM E=17, d=157, n=2773  from  (RSA)
2030 INPUT"Encryption E=<17> ";E
2040 INPUT"Decryption D=<157>";D
2050 INPUT"Composite Number N=<2773> ";N
2060 PRINT:PRINT"*** アンゴウ サクセイ ***"
```

図5

```
2070 PRINT:PRINT"m(i)          c(i) "
2080 FOR J=1 TO W
2090 M=VAL(M$(J))
2100 C=M
2110 FOR I=1 TO E-1
2120 C=FNMF(C*M)
2130 NEXT I
2140 C(J)=C
2150 '
2160 PRINT USING" ####            ####";M,C
2170 NEXTJ
2180 PRINT:GOSUB 4500
2190 RETURN
2500 PRINT CHR$(12):PRINT:PRINT"*** ﾌｸｺﾞｳ ***"
2510 PRINT:PRINT " C(I)          M(I)"
2520 FOR J=1 TO W
2530 P=C(J)
2540   FOR I=1 TO D-1
2550   P=FNMF(P*C(J))
2560   NEXTI
2570 '
2580 PRINT USING" ####            ####":C(J),P
2590 M(J)=P
2600 NEXTJ
2610 PRINT :GOSUB 4500
2620 RETURN
3000 '"ｼﾞｼﾝﾌﾞﾝ ｼｭﾂﾘｮｸ"
3010 PRINT"ｼﾞｭｼﾝﾌﾞﾝ ﾊ ﾂｷﾞﾉ ﾄｳﾘﾃﾞｽ":PRINT
3020 FOR I=1 TO W
3030 MR$=STR$(M(I))
3040 LR=LEN(MR$)
3050 IF LR>=5 THEN K=3:GOTO 3090
3060 IF LR=3 THEN MID$(MR$,1,1)="0":K=1:GOTO 3090
3070 IF LR=4 THEN MID$(MR$,1,1)="0":K=2:GOTO 3090
3080 IF LR=4 THEN MID$(MR$,1,1)="0":MR$="00"+MR$:K=2
3090 L$=LEFT$(MR$,K):R$=RIGHT$(MR$,2)
3100 F=VAL(L$)+64:B=VAL(R$)+64
3110 IF F=64 THEN F=32 ELSE IF B=64 THEN B=32
3120 '
3130 PRINT CHR$(F);CHR$(B);
3140 NEXT I
3150 GOSUB 4500
3160 RETURN
4000 REM  "ｾﾞﾝﾌﾞﾝ ｼﾂﾘｮｸ"
4010 '
4020 '
4030 '
4040 PRINT CHR$(12):PRINT "ｾﾞﾝﾌﾞﾝ ｼﾂﾘｮｸ" :PRINT
4050 PRINT "     ﾊｯｼﾝﾌﾞﾝ ﾊ":PRINT
4060 PRINT "/":M$;"/";" ";L;"ﾓｼﾞﾃﾞｽ":PRINT
4070 PRINT "ｺｰﾄﾞ  ｱﾝｺﾞｳ   ﾌｸｺﾞｳ "
4080 FOR J=1 TO W
4090 '
4100 PRINTM$(J);
4110 PRINTUSING"  #####   #####";C(J),M(J)
4120 NEXTJ
4130 PRINT:GOSUB3010
4140 RETURN
4500 REM "ｳｴｲﾄ"
4510 PRINT:PRINT:PRINT"ﾊｼﾞﾒﾆ ﾓﾄﾞﾙﾆﾊ Space Key ｦ ｵｼﾃｸﾀﾞｻｲ"
4520 A$=INKEY$:IF A$<>" " THEN GOTO 4520
4530 RETURN
5000 END
```

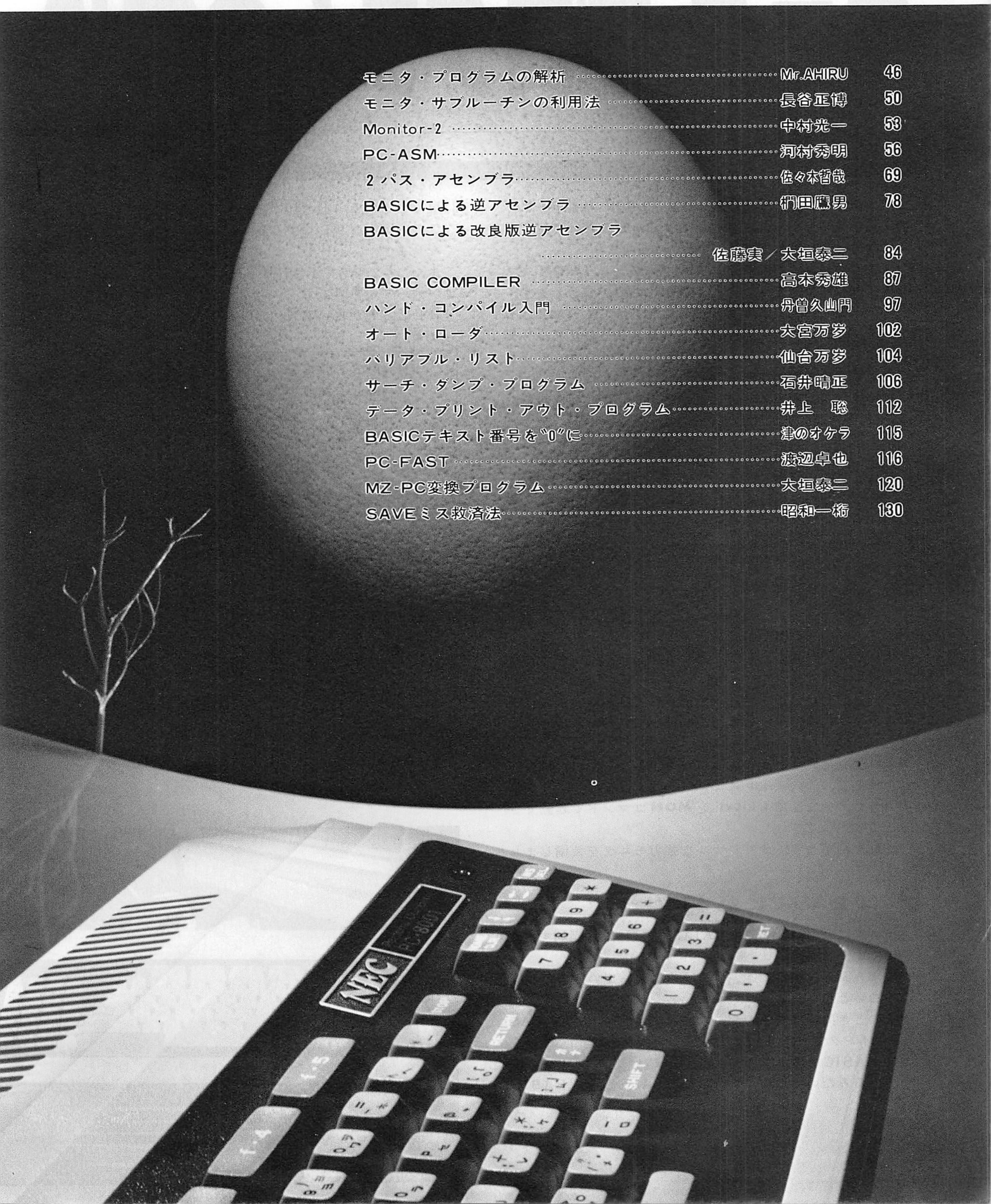
図3

```
40 REM  ***zaiko****bv k.kawai
50 REM  ***ｼｲﾚｰｳﾘｱｹﾞｰｻﾞｲｺｰﾄｲｱﾜｾｰｳﾘｱｹﾞｸﾞﾗﾌ***
60 X=10:DIM N(X),T(X),S(X),Y#(X)
70 Y=10:DIM US(Y),D#(Y),T#(50),M(50)
80 N=0:G#=0::P=1:Z#=0
100 K=10:W=0
110 DIM N$(K),ST(K),SS(K),SD#(K)
120 GOSUB 9000
200 PRINT CHR$(12)
210 PRINT "***  ﾌﾟﾛｸﾞﾗﾑ  ***"
220 PRINT
230 PRINT "1,ｼｲﾚ"
240 PRINT "2,ﾃﾞﾝﾋﾟｮｳ"
250 PRINT "3,ｳﾘｱｹﾞ"
260 PRINT "4,ｻﾞｲｺ"
270 PRINT "5,ﾄｲｱﾜｾ"
280 PRINT "6,ｵﾜﾘ"
300 INPUT "ﾄﾞﾚﾆｼﾏｽｶ   1-6";A
310 ON A GOSUB 1000,2000,3000,4000,5000,6000
320 GOTO 200
1000 REM **** ｼｲﾚ****
1150 PRINT CHR$(12):PRINT "ｼｲﾚｻｷﾞｮｳ":PRINT
1160 INPUT "ｺｰﾄﾞ";N
1170 IF N=999 THEN 1500
1180 INPUT"ｼｲﾚﾀﾝｶ";ST(N)
1190 INPUT"ｼｲﾚｽｳﾘｮｳ";S:PRINT
1200 PRINT N;"/";N$(N);"/";ST(N);"/";S:PRINT
1210 INPUT"ｶｸﾆﾝ  ﾀﾀﾞｼｲ y/ﾏﾁｶﾞｲ n";A$
1220 IF A$="n" GOTO 1160
1230 SS(N)=SS(N)+S
1240 Y#=ST(N)*S
1250 SD#(N)=SD#(N)+Y#
1260 GOTO 1150
1500 REM ***  ｼｲﾚﾌｧｲﾙ ***
1510 SZ#=0
1520 PRINT"       ｼｲﾚ ﾌｧｲﾙ ｲﾁﾗﾝﾋｮｳ":PRINT
1530 PRINT"ｺｰﾄﾞ   ﾋﾝﾒｲ      ﾀﾝｶ     ｽｳﾘｮｳ          ｷﾝｶﾞｸ"
1540 PRINT
1550 FOR X=1 TO 10
1560 PRINT USING"##     ";X;
1570 PRINT N$(X);TAB(14);
1580 PRINT USING "####    ####     ##########";ST(X),SS(X),SD#(X)
1590 SZ#=SZ#+SD#(X)
1600 NEXT X
1610 PRINT"----------------------------------------"
1620 PRINT TAB(6) "ｺﾞｳｹｲ";TAB(26);
1630 PRINT USING"¥¥#########,";SZ#
1640 PRINT:INPUT"ｽﾐﾏｼﾀﾗ ｷｰｲﾝ";B$
1650 RETURN
1999 REM **********************
2000 REM **    ﾃﾞﾝﾋﾟｮｳ ｻｷﾞｮｳ   **
2010 T#=0:I=0
2020 I=I+1:PRINT CHR$(12):PRINT"          ﾃﾞﾝﾋﾟｮｳ ﾊｯｺｳ    ":PRINT
2030 INPUT"ｺｰﾄﾞﾊ";N(I)
2040 INPUT"ﾀﾝｶﾊ";T(I)
2050 INPUT"ｽｳﾘｮｳﾊ";S(I)
```

図3

```
2060 Y#(I)=T(I)*S(I)
2070 N=N(I)
2075 UT(N)=T(I)
2080 US(N)=US(N)+S(I)
2090 D#(N)=D#(N)+Y#(I)
2100 IF I=5 GOTO 2020
2110 INPUT"ｼｮｳｹｲ ﾅﾗ (*)ｦｲﾚﾙ";A$
2120 IF A$<>"*" GOTO 2020
2130 PRINT CHR$(12):PRINT"no.";P:PRINT
2140 PRINT"ｺｰﾄﾞ";"ﾋﾝﾒｲ";SPC(3);"ｽｳﾘｮｳ";SPC(7);"ｷﾝｶﾞｸ"
2150 FOR N=1 TO I
2160 A=N(N)
2170 PRINT USING"  ####    ####     #########";T(N),S(N),Y#(N)
2200 T#=T#+Y#(N)
2210 NEXT N
2220 PRINT"----------------------------------------"
2230 PRINT TAB(25):PRINT USING"¥¥#########,";G#
2240 G#=G#+T#:P=P+1
2250 PRINT: INPUT"ｺﾞｳｹｲ ﾅﾗ (¥)ｦｲﾚﾙ";A$
2260 IF A$="¥" GOTO 2280
2270 M(P)=I:T#(P)=T#:GOTO 2010
2280 PRINT:PRINT"ﾏｲﾄﾞ ｱﾘｶﾞﾄｳ ｺﾞｻﾞｲﾏｽ";SPC(6)
2290 PRINT USING"¥¥#########,";G#
2300 Z#=Z#+G#:G#=0
2310 INPUT"ｽﾐﾏｼﾀﾗ ｷｰｲﾝ";T$
2320 RETURN
2990 REM **********************
3000 PRINT CHR$(12):PRINT"          ***  ｳﾘｱｹﾞｲﾁﾗﾝﾋｮｳ  ***":PRINT
3010 PRINT"ｺｰﾄﾞ";" ﾋﾝﾒｲ";SPC(3);"ﾀﾝｶ";SPC(3);"ｽｳﾘｮｳ";SPC(8);"ｷﾝｶﾞｸ":PRINT
3020 FOR I=1 TO 10
3030 PRINT USING"##  ";I;
3040 PRINT N$(I);TAB(11)
3050 PRINT USING"####     ####     #########";UT(I),US(I),D#(I)
3060 NEXT I
3070 PRINT"----------------------------------------"
3080 PRINT SPC(25);
3090 PRINT USING"¥¥#########,";Z#
3100 INPUT "ｽﾐﾏｼﾀﾗ ｷｰｲﾝ";Q$
3110 RETURN
4000 REM *** ｻﾞｲｺ ｲﾁﾗﾝﾋｮｳ ***
4010 PRINT CHR$(12):ZZ#=0
4020 PRINT "        *** ｻﾞｲｺ ｲﾁﾗﾝﾋｮｳ ***"
4030 PRINT
4040 PRINT"ｺｰﾄﾞ  ﾋﾝﾒｲ      ﾀﾝｶ    ｽｳﾘｮｳ          ｷﾝｶﾞｸ"
4050 PRINT
4060 FOR J=1 TO 10
4070 ZS(J)=SS(J)-US(J)
4080 ZD#(J)=ZS(J)*ST(J)
4090 ZZ#=ZZ#+ZD#(J)
4100 PRINT USING"##     ";J;
4110 PRINT N$(J);TAB(14);
4120 PRINT USING"####    ####    ########,";ST(J),ZS(J),ZD#(J)
4130 NEXT J
4140 PRINT "----------------------------------------"
4150 PRINT SPC(26);
4160 PRINT USING"¥¥#########,";ZZ#
4170 INPUT "ｽﾐﾏｼﾀﾗ ｷｰｲﾝﾌ";B$
4180 RETURN
5000 REM *** ﾌｧｲﾙ ﾓﾆﾀ ***
5010 PRINT CHR$(12)
5020 PRINT"***** ﾌｧｲﾙ ﾓﾆﾀ *****"
5030 PRINT
5040 PRINT "     ｼｲﾚ ﾌｧｲﾙ (1)"
5050 PRINT "     ｳﾘｱｹﾞﾌｧｲﾙ(2)"
5060 PRINT "     ｻﾞｲｺﾌｧｲﾙ (3)"
5070 PRINT
5080 INPUT"(1)-(3) ﾉ ﾄﾞﾚﾃﾞｽｶ";A
5090 PRINT CHR$(12)
5100 ON A GOSUB 1500,3000,4000
5110 RETURN
6000 PRINT" ｵﾜﾘ ":END
9000 REM **    ｼｮｳﾋﾝﾒｲ ﾉ ﾖﾐｺﾐ   **
9010 FOR I=1 TO10
9020 READ N$(I)
9030 NEXT I
9040 RESTORE:RETURN
9050 DATA"ｲﾁｺﾞ"
9060 DATA"ﾌﾞﾄﾞｳ"
9070 DATA"ｽｲｶ"
9080 DATA"ﾘﾝｺﾞ"
9090 DATA"ﾐｶﾝ"
9100 DATA"ﾊﾞﾅﾅ"
9110 DATA"ﾅｼ"
9120 DATA"ﾓﾓ"
9130 DATA"ｶｷ"
9140 DATA"ﾒﾛﾝ"
```

システム・プログラム編



0.9Kバイトに収められているモニタ内のサブルーチンを活用しよう！

モニタ・プログラムの解析

PC-8001

Mr. AHIRU

PC-8001は，相変わらず人気がありますね．私もユーザーの１人です．しかし，あのハードをBASICだけで使用するのはもったいないのです．

そんなわけで，ＰＣのモニタ・プログラム（**MON**コマンドで起動するモニタのこと）を調べたので報告します．

PC-8001ではI/Oのサブルーチンなどは，N-BASICインタープリタの中に入っており，モニタはそのサブルーチンをコールして使っています．

これは通常のプログラムからみれば逆ですね．

モニタ・プログラムの開始番地は，５Ｃ２Ｃ番地からで，5FFF番地までを使用しています．

モニタについて一言

1 モニタを強化したい方

モニタ・プログラムは約0.9Kバイトの容量なので，どうしても機能不足になってしまいます．

このモニタはスタート時に７ＦＦＦ番地の内容（このアドレスは４個目のＲＯＭの最後のところ）をチェックし，その内容が55Ｈであれば７ＦＦＣ番地にジャンプするようになっています．

４個目のＲＯＭは，現在，販売されていませんが，拡張できるようになっているのは，間違いないのでそのうち販売されるのかな？

それはさておき，現在販売されているPC-8011，PC-8012を使って，このアドレスにPROMを入れ，そのプログラム（モニタでなくてもいい）を**MON**コマンドで起動できるのです．

これを利用して，オリジナルな強力モニタを装備しましょう．

2 N-BASICに戻るには…

モニタはN-BASICからCALLで呼ばれて，起動するみたいです．

N-BASICに戻るとき、リターンで戻っているので，間違いないと思います．

そのとき問題になるのが，スタックとテキスト(N-BASIC)ポインタです．

N-BASICではテキスト・ポインタはＨＬレジスタを使っているみたいで，モニタの最初でSAVEし，N-BASICに戻るときにセットされています．

この内容をおかしくすると，BASICに戻ったときおかしくなります．

また，スタックで示されるメモリにはBASICへの戻り番地が入っているのです．

ＨＬとＳＰがSAVEされている番地は次のようになっています．

ＨＬ➡ＦＦ34番地
ＳＰ➡ＦＦ36番地

これを利用すると，マシン語プログラムから直接BASICに戻れるはずです．

ＨＬとＳＰをセットしてＲＥＴすればいいのです．

3 モニタの再スタート

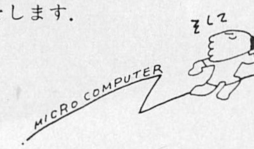
モニタによって，マシン語をメモリに書き込むのはいいのですが，モニタへ戻せないと，非常に不便な場合があります．

その場合，５Ｃ66番地にジャンプすれば，スタックを再セットし，コマンド入力待ちになります．

また，５Ｃ５Ｅ番地にすれば，ＣＲ/ＬＦ後に，“？”を表示しコマンド入力待ちになります．

＊　＊　＊

次に，サブルーチンの説明をします．





サブルーチンの内容

1 スペース表示

アドレス：5FD4番地
入力パラメータ：なし
出力パラメータ：なし
機能：画面にスペースを表示します．

2 CR/LF出力

アドレス：5FCA番地
入力パラメータ：なし
出力パラメータ：なし
機能：画面にCR/LFを出力，すなわち復改を行ないます．

3 1文字表示&カーソル表示

アドレス：5FB0番地
入力パラメータ：Accに文字コード
出力パラメータ：なし
機能：Accの内容を画面に表示し，カーソルも表示します．

4 ペア・レジスタの比較

アドレス：5ED3番地
入力パラメータ：HLとDEレジスタに比較したい値をセット
出力パラメータ：″C″＝1➡HL<DE
″Z″＝1➡HL＝DE
機能：HLレジスタとDEレジスタを比較し，HLレジスタの方が大きければキャリーフラグがセットされます．また，等しければゼロ・フラグがセットされます．

5 コード変換(BIN→HEX)&表示

アドレス：5EC5番地
入力パラメータ：Accにバイナリ・コード
出力パラメータ：なし
機能：Accのバイナリ・コードをHEXコードに変換し，そのコードを画面に表示します．

6 HLレジスタの内容を表示

アドレス：5EC0番地
入力パラメータ：HLレジスタにバイナリ・コード
出力パラメータ：なし
機能：HLレジスタにセットされているバイナリ・コードをHEXコードに変換して画面に表示します．

7 メモリの内容を表示

アドレス：5EBD番地
入力パラメータ：表示したいメモリの番地をHLレジスタにセット
出力パラメータ：なし
機能：HLレジスタで示される番地の内容（バイナリ）を画面に表示します．

8 コード変換(HEX→BIN)

アドレス：5EA0番地
入力パラメータ：DEレジスタにHEXコード
出力パラメータ：Accにバイナリ・コード
機能：DEレジスタにセットされているHEXコードをバイナリ・コードに変換しAccにセットします．

9 コード変換(BIN→HEX)

アドレス：5E83番地
入力パラメータ：Accにバイナリ・コード
出力パラメータ：DEレジスタにHEXコード
機能：Accにセットされているバイナリ・コードをHEXコードに変換し，DEレジスタにセットします．

10 HEXコード・チェック

アドレス：5E39番地
入力パラメータ：Accにコード
出力パラメータ：″C″＝1➡NOT
機能：Accのコードをチェックして，それがHEXコードかどうかチェックし，その結果がキャリーフラグにセットされます．

* * *

以上が，モニタ・プログラムの中のサブルーチンの1部ですが，次にモニタが使用しているN-BASICインタープリタのサブルーチンを説明します．

11 1文字画面表示

アドレス：257番地
入力パラメータ：Accに出力する文字コード
出力パラメータ：なし
機能：Accにセットされているコードに対応するキャラクタを画面に表示します．

このサブルーチンで表示される文字の位置は，現在のカーソルのある位置に行なわれ，カーソルは次に進みます．

PC-8001のキャラクタ・コードは表1のようになっています．

このサブルーチンを使用せずに，直接メモリにストアする場合は，現在の画面が80文字or40文字のどちらになっているか知る必要があります．

80文字の場合は，F300番地（V-RAM）から順番にストアすればいいのですが，40文字の場合は，F300番地の次にF302番地といった具合に1つおきになるからです．

いまの画面の状態は，EA65番地の内容をチェックすればわかります．

12 キー入力

アドレス：F75番地
入力パラメータ：なし
出力パラメータ：Accに文字コード
機能：キーが入力されればAccに文字コードがセットされて，リターンします．

13 STOPキーチェック

アドレス：CF1番地
入力パラメータ：なし
出力パラメータ：″C″＝1➡STOP KEY ON
機能：STOPキーが押されているかどうかをチェックし，押されていれば，キャリーフラグがセットされます．押されていなければ，キャリーフラグ＝0でリターンします．

表1　PC-8001　キャラクタ・コード表

下位＼上位	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		D_E		0	@	P		p	▁	┴		ー	タ	ミ	═	×
1	S_H	D_1	!	1	A	Q	a	q	▂	┬	。	ア	チ	ム	╞	円
2	S_X	D_2	"	2	B	R	b	r	▃	┤	「	イ	ツ	メ	╪	年
3	E_X	D_3	#	3	C	S	c	s	▄	├	」	ウ	テ	モ	╡	月
4	E_T	D_4	$	4	D	T	d	t	▅	▔	、	エ	ト	ヤ	◢	日
5	E_Q	N_K	%	5	E	U	e	u	▆	─	・	オ	ナ	ユ	◣	時
6	A_K	S_N	&	6	F	V	f	v	▇	│	ヲ	カ	ニ	ヨ	◥	分
7	B_L	E_B	'	7	G	W	g	w	█	▕	ァ	キ	ヌ	ラ	◤	秒
8	B_S	C_N	(	8	H	X	h	x	▏	┌	ィ	ク	ネ	リ	♠	
9	H_T	E_M	)	9	I	Y	i	y	▎	┐	ゥ	ケ	ノ	ル	♥	
A	L_F	S_B	*	:	J	Z	j	z	▍	└	ェ	コ	ハ	レ	♦	
B	H_M	E_C	+	;	K	[	k	¦	▌	┘	ォ	サ	ヒ	ロ	♣	
C	C_L	→	,	<	L	¥	l	:	▋	╭	ャ	シ	フ	ワ	●	
D	C_R	←	-	=	M	]	m	¦	▊	╮	ュ	ス	ヘ	ン	○	
E	S_O	↑	.	>	N	^	n	~	▉	╰	ョ	セ	ホ	゛	╱	
F	S_I	↓	/	?	O	_	o		┼	╯	ッ	ソ	マ	゜	╲	

14　カーソル表示

アドレス：BE2番地
入力パラメータ：なし
出力パラメータ：なし
機能：カーソルの表示を行ないます.

15　カーソルの消去

アドレス：BD2番地
入力パラメータ：なし
出力パラメータ：なし
機能：カーソルの消去

16　キー入力チェック

アドレス：F7B番地
入力パラメータ：なし
出力パラメータ："C"＝0➡キーON➡Acc＝文字コード
機能：キー入力をチェックし，キーが入力されていれば，キャリーフラグをリセットし，Accに文字コードをセットしてリターンします.

注意

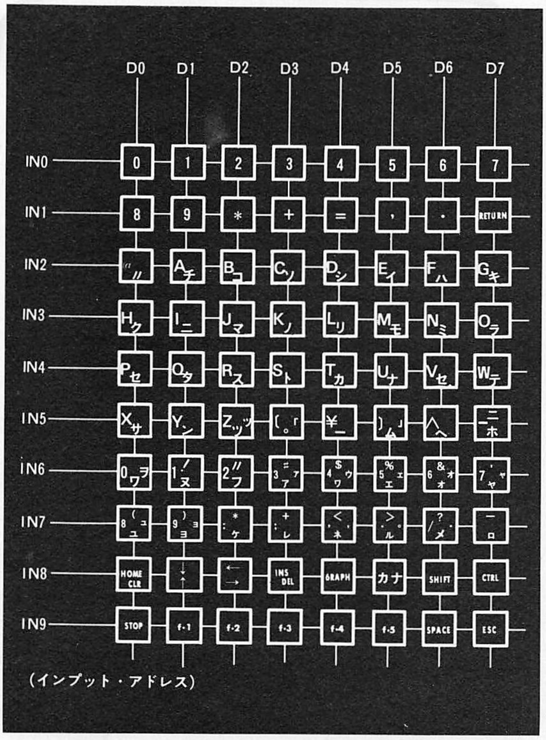
画面に文字を連続して表示する場合は，カーソルを消してから行なった方が速くなります．試してみてください.

以上で，使えるサブルーチンは終わりですが，物足りない方が多いと思うので，画面の初期設定，プリンタの制御などの例について説明したいと思います.



淋しい港町でおました…

図1　キーマトリックス



リスト1　プリンタ1文字出力サブルーチン

ラベル	ニモニック	オペランド	コメント
LPOUT	PUSH	AF	
LPOT1	IN	A, 40H	
	AND	1	
	JR	NZ, LPOT1	Ready?
	POP	AF	
	OUT	10H, A	データ・セット
	PUSH	AF	
	LD	A, (0EA67H)	
	AND	0FEH	
	OUT	40H, A	ストローブ・セット
	OR	1	
	OUT	40H, A	ストローブ・リセット
	LD	(0EA67H), A	
	POP	AF	
	RET		

注）Acc＝出力する文字コード

参考までに

1　キー入力

先ほどのサブルーチンを使用しないで，キー入力プログラムを作る場合，IN命令で行なうことができますが，そのときのアドレスは0～9で図1に示すように対応しています.

キーが押されていれば，それに対応したビットが"0"になります.

また，チャタリング・ウェイト時間は5～8msでいいみたいです.

2　プリンタ出力

リスト1に示すプログラム例を参考にしてください.

出力ポートを制御する場合，ポートがいろんな制御に使用されている関係上，ポートの状態をメモリにストアしておく必要があり，その内容に対して，OR，ANDした値

リスト2　画面の初期設定

ニモニック	オペランド	コメント
LD	A,(0EA66H)	内容 76543210
OR	3	カラー "0"→カラー "1"→白黒　40/80 "0"→40文字 "1"→80文字
OUT	30H,A	
LD	(0EA66H),A	
LD	A,(0EA67H)	
OR	8	
OUT	40H,A	→80文字で白黒の場合
LD	(0EA67H),A	
Ⓐ XRA	A	ストップ DISPLAY
OUT	51H,A	
LD	A,80H	
OUT	68H,A	
LD	A,0	V-RAM
OUT	64H,A	アドレス(F300)セット
LD	A,0F3H	
OUT	64H,A	
LD	A,5FH	→25行の場合0B7H
OUT	65H,A	カウンタ(20行)セット
LD	A,89H	
OUT	65H,A	→25行の場合8BH
LD	A,0CEH	
OUT	50H,A	
LD	A,93H	→25行の場合98H
OUT	50H,A	
LD	A,69H	→25行の場合67H
OUT	50H,A	
LD	A,0BEH	→25行の場合0DEH
OUT	50H,A	
LD	A,13H	→カラーの場合53H
OUT	50H,A	
LD	A,43H	
OUT	51H,A	
LD	A,0C4H	DMA イネーブル
OUT	68H,A	
LD	A 20H	スタート DISPLAY
OUT	51H,A	
IN	A,40H	VRTC チェック
AND	20H	
JR	NZ,$-4	
IN	A,40H	
AND	20H	
JR	Z,$-4	
Ⓐと同じ		(OUT 64H,65H,68HはDMAコントローラ OUT 50H,51HはCRTコントローラ)
IN	A,40H	
AND	20H	
JR	NZ,$-4	
LD	A,(0EA67H)	
AND	0F7H	
OUT	40H,A	
LD	(0EA67H),A	

をホートにセットして制御します.

> ＥＡ66番地→30Ｈの出力ポートの内容
> ＥＡ67番地→40Ｈの出力ポートの内容

プリンタの場合，**IN A,40H**でReadyをチェックし，Readyであれば，**OUT 10H,A**でデータをセットし，その後

リスト3　カーソルの表示

ニモニック	オペランド	コメント
IN	A,40H	VRTC チェック
AND	20H	
JR	NZ,$-4	
IN	A,40H	
AND	20H	
JR	Z,$-4	
LD	A,81H	→カーソルを消す場合は80H
OUT	51H,A	
LD	A,XXH	→文字数－2
OUT	50H,A	
LD	A,YYH	→行数－1
OUT	50H,A	

OUT 40H,Aでプリンタへのストローブのセット/リセットを行なえば１文字プリントすることができます.

3 画面の初期設定

リスト2に示すプログラム例を参照してください.

画面の初期設定を行なうには，ポートの設定，ＤＭＡコントローラの設定，ＣＲＴコントローラの設定などを行ないます.

細かい説明はしませんが，例のとおり行なえば初期設定ができます.

4 カーソルの表示

先に説明した，インタープリタのカーソル表示サブルーチンは，カーソルを表示するだけですが，ここではポジションを指定する場合のＣＲＴコントローラの制御例について説明します.

ＣＲＴコントローラのVRTCをチェックし，コマンド・セット，パラメータ・セットを行なうことによって，指定した位置にカーソルを表示できます.

また，カーソルを消す場合も同様に行ないますが，この場合のポジションは別に意味を持ちません.

それでは，**リスト3**を参考にしてください.

最後に

今回の説明は一例なので，もっと良い使い方があるかもわかりませんが，参考にしていただければ幸いです.



PC-8001 モニタ・サブルーチンの利用法

●長谷正博●

PC-8001はモニタ・プログラムが公開されていないので，機械語だけのプログラムを作るとき，特に画面制御が難しく，どうしてもBASICとの混用をしてしまいます。

そこで，モニタ・プログラムの解読を進め，利用できそうなサブルーチンなどをリスト・アップしてみたので，この利用方法を考えてみたいと思います。

なお，ルーチンのラベルは適当に付けたものです。

＊モニタ関係

BASICから MON コマンドで機械語モニタに移ります．ここで利用できそうなルーチンを示します。

1 MONJP $5C66 (JUMP TO MONITOR)

G コマンドで機械語プログラムを走らせることはできますが，これをストップさせる適当な命令がありません．HLT を使ったのでは後の操作がやっかいです。

そこで，プログラムの終了はこのアドレスにジャンプさせれば再びモニタのコマンド受け付け状態になるので便利です。

さらに，$5C5Eでは？を表示してモニタに移るので，エラーを検出して戻る場合などに利用できます。

2 CTRLB (CONTROL B, ↑B)

```
LD HL, $FF34
JP $41FA
```

この操作では画面がクリアされないまま OK 表示をします．BACIC に戻ったわけです。

ここで，S コマンドの操作について1つ．アクセスするアドレスを1番地戻すには，説明書では ⇧H を押すことになっていますが，これを INS/DEL キーでも行なえます．片手でできるので便利です。

余談になりますが BASIC 中で ⇧H を行なえば1文字消すことができます。

もう1つ余談ですが，前の G コマンドでアドレスを入力しないままキャリッジ・リターンすると，HLレジスタに$0000が入ったままで，これがジャンプ・アドレスとなり，大変なことになります．注意しましょう。

3 NEW (BASIC NEW)

```
XOR A
JP $3DE0
```

4 RUN $3DF4 (BASIC RUN)

```
*G3DF4 (MONより)
またはJP $3DF4
```

これで機械語から BASIC プログラムが直接走ります．自動化の手掛りになるのではないでしょうか。

5 LIST $570C (BASIC LIST)

```
*G570C
またはJP $570C
```

機械語から BASIC のリストを出力しますが，BASICに変わってしまいます。

＊ディスプレイ関係

1 CONSOL 0884 (BASIC CONSOLE)

BASIC の CONSOLE ダイレクト・コマンドが使えるようにしたものです。

まず，HL レジスタにCONSOLEのステイタス・ポインタを設定します．ステイタス・バッファには次のように記入しておきます。

```
CONBUF 302C32352C312C3100
```

これは CONSOLE 0,25,1,1 と設定したことになります．終わりの"00"を必ず入れてください。

```
LD HL, CONBUF
CALL CONSOL
```

省略型も試みてください。

2 WIDTH $0843 (BASIC WIDTH)

CONSOL と同じように，ステイタス・バッファを作ります．【例】38302C323500＝WIDTH80, 25

```
LD HL, WIDBUF
CALL WIDTH
```

この WIDTH 設定法には縦と横を別々にセットする方法もあります。

```
HWSET $09A3
```

```
LD A, HW
CALL HWSET
```

Aレジスタの値はHW40のとき$28，HW80のとき$50です。

```
VWSET $09D7
```

```
LD A, VW
CALL VWSET
```

Aレジスタの値はVW20のとき$14，HW25のとき$19です。

3 COLOR $0951 (BASIC COLOR)

前記**1**，**2**と同じように設定します(【例】332C33322C3000).

```
LD HL, COLBUF
CALL COLOR
```

以上のような方法でLOCATEもできます.

4 FKYDSP $07C9 (FUNCTION KEY DISPLAY)

ファンクション・キーの表示を行ないます．Aレジスタには通常〝00″または〝05″を記入します.〝00 では1～5まで,〝05″では6～10のキーを表示します.

〝01″と入れると2～6を表示します.

このルーチンはファンクション・キーに別の表示をさせることが可能であることを示し，表1のキーアドレスからキャラクタ・コードを記入しておきます.

```
LD A, $00($05)
CALL FKYDSP
```

5 FKYOFF $0B2E (FKEY OFF)

```
CALL FKYOFF
```

この方法ではキー表示を消しますが，スクロール範囲は元のままです.

6 CUSADS $03F3 (CURSOR ADDRESS)

Aレジスタに指定したカーソル位置を実際のアドレスに直します．結果はDEレジスタです.

```
CALL CUSADS
```

表1　ファンクション・キーとタイマ関係のアドレスとデータ

アドレス	データ	
$EA76	SS	SECOND
77	MM	MINUTE
78	HH	HOUR
79	DD	DAY
7A	MT MT	MONTH
7B	YY	YEAR
7C～	ファンクション・キー	1
8C～	〃	2
9C～	〃	3
AC～	〃	4
BC～	〃	5
CC～	〃	6
DC～	〃	7
EC～	〃	8
FC～	〃	9
$EB0C～	〃	10

7 CLRLIN $0451 (CLEAR LINE)

DEレジスタに消去する行の先頭アドレスを記入します．次の例は4行消去する場合です.

```
       LD A, $04
       LD DE, $FC60
CLRLP: CALL CLRLIN
       DEC A
       JPNZ CLRLP
```

8 CR & LF $5FCA (RETURN & LINE FEED)

CRとLFを1つにしたルーチンです.

＊　＊　＊

ここからは前記のルーチンなどを利用して作った自作のルーチンです.

9 FKYINT (FKEY INITIALIZE)

ファンクション・キーに別のキャラクタを設定した場合に，これを元のBASIC時の状態に戻します．機械語プログラム終了直前などに利用できます.

```
LD HL, $3A81
LD DE, $EA7C
LD BC, $0050
LDIR
XOR A
CALL FKYDSP
RET
```

10 SQSET (SCROLL SET)

現在のカーソル位置から下をスクロール範囲に設定します．スクロールが行なわれても，ある表示を残しておきたいときに利用できます.

```
CALL CR & LF
LD HL, $EA63
LD A, L
LD ($EA5E), A
LD B, A
LD A, ($EA62)
SUB B
LD ($EA5D), A
RET
```

表2　カラー指定とコードの関係

カラー		マシン・コード	
		キャラクタ	グラフィック
ブラック	0	08	18
ブルー	1	28	38
レッド	2	48	58
バイオレット	3	68	78
グリーン	4	88	98
パール・ブルー	5	A8	B8
イエロー	6	C8	D8
ホワイト	7	E8	F8

ウインド・スクロールも考えてください.

11 SQCLR (SCROLL CLEAR)

モニタ・プログラムのホーム・クリアは全画面を消去してしまい，残しておきたい表示も消えてしまいます．このプログラムはスクロール範囲のみを消去します.

```
       LD A, $EA5E
       PUSH AF
       DCR A
       CALL CUSADS
       POP AF
       LD A, ($EA62)
       SUB B
       LD B, A
CLRLP: CALL CLRLIN
       DEC A
       JPNZ CLRLP
       LD A, $0B
       CALL CONOUT($0257)
       RET
```

ウインド・スクロールの場合は多少の変更が必要です.

＊カラー化のヒント

本誌にもPC-8001による色をほどこした機械語ゲーム(BASIC混用)が発表されていますが，機械語で個々の表示物に固有の色を彩け，これを移動させるとなると相当に難しいようです.

ここで，機械語によるカラー化のヒントになるプログラム例を紹介します.

画面の位置設定はカーソルで行ないますが，実際の現カーソル位置とは関係ありません．この方法ではカラー時に横幅が1つ減り，左端の縦1列が使えませんが，横幅の設定(WIDTH)に関係なく利用できます.

```
LD HL, $0101       カーソル法による画面の位置設定の例
PUSH HL
LD A, $78          カラー指定の一例
CALL ATTRIB        ATTRIB $04F8
POP HL
```

```
CALL CUSADS  実際のアドレスに変換
LD A, $35    表示キャラクタ例
LD (HL), A   キャラクタ表示
JP MONIJP    モニタに戻る
```

この例ではカーソル位置（01, 1）を紫色でグラフィックとし，コード$35のグラフィック・キャラクタを表示します．カラー指定を$68にすればキャラクタで5と同色で表示します．カラーの指定は表2を参考にしてください．

このカラーコードの下位桁が8以外のときにも種々な表示がされますが，これについてはまだ何も調べていません．

＊その他のサブルーチン

1 AZLCNV $5FC1 (A～Z LARGE CONVERT)

A～Zまでの英小文字（61～7A）を大文字（41～5A）に変換します．Aレジスタが入出力になっています．

2 HEXCHK $5E39 (HEX. CHECK)

16進に使用されるキャラクタかどうか調べます．それ以外のものはキャリーフラグが設定されます．

3 BINCV4 $5E4B (BINARY CONVERT 4)

4桁までの16進ASCIIコードをバイナリ・コードに変換します．Aレジスタが入力になりますが1度に1桁の変換しかなされないので，ループで廻し，セパレータで完了します．以下に，前述の1,2を利用した変換方法を示します．

```
      LD DE, ASCBUF
      LD HL, $0000
BINLP: LD A, (DE)
      INC DE
      CALL AZLCNV
      CALL HEXCHK
      JP C, NEXT
      CALL ASBIN4
      JR BINLP
NEXT: LD BUF, HL
```

このルーチンでは何桁変換してもかまいませんが，セパレータを検出する直前の4桁が Lレジスタに残ります

4 BINCV2 $5EA0 (BINARY CONVERT 2)

DEレジスタ中の2桁のASCIIコードをバイナリ・コードに変換します．結果はAレジスタに残ります．

5 ASCDP4 $5EC0 (ASCII CONVERT & DISPLAY 4)

HLレジスタ中の4桁のバイナリ・コードをASCIIコードに変換し，カーソル位置から表示します．

6 ASCDP2 $5EBD (ASCII CONVERT & DISPLAY 2)

HLレジスタをポインタとするメモリの内容を2桁のASCIIコードに変換し，カーソル位置で表示します．

7 SBCHD $5ED3 (SBC HL, DE)

HL-DEの減算を行ないますが，結果はフラグのみで，HLレジスタの値は元のままです．

＊タイマについて

PC-8001は内部にハードウェア・クロックを持ち，その時刻，日付はTIME$，DATE$で読み書きできますが，これらを機械語で行なうのが次の3つのサブルーチンです．

1 TMREAD $1602 (TIME READ)

結果は表1のアドレスに入ります．

表現方法は各1バイトでそれぞれ上位桁が×10の位を示します．たとえば，$EA76が23となっていれば23秒の意味です．

YYは実際はこのクロックから読み出されたものでなく，初めにセットされたものが残っているだけで，1年経っても繰り上がりはしません．

2 TMSET $1663 (TIME SET)

表1の各アドレスに時刻および日付（YYも）をセットしてこのルーチンを呼びます．

3 TMCLR $164C (TIME CLEAR)

タイマを初期状態にします（00:00:00 01/01/79)．これらのルーチンはシーケンシャルなコントロールを行なうプログラムに活用できると思います．

おわりに

これら紹介したルーチンはDISK BASICでは動作確認をしていないので，ご了承願います．これらはほんの一部の解読でしかありません．まだまだ便利なルーチンがあると思います．

◘参考文献

1) 栩田鷹男：“BASICによる逆アセンブラ”，I/O，'80年2月号
2) 石井晴正：“PC-8001サーチダンププログラム”，I/O，'80年3月号
3) 田中二郎：“機械語開発ツール”，ASCII，1979/12
4) N-BASIC入門，アスキーラボラトリーズ

スクリーン・エディット・チェック・サム機能の付いたPC用モニタ

Monitor-2

マシン語プログラムの入力，デバッグに便利！！

中村光一

高校に入ってからマイコンを知り，１ヶ月ほど前にＰＣ-8001を手に入れました．そして，I/Oに載っているゲームをかたっぱしから入れました．しかし，ＰＣのモニタは機能が少なく，アドレスの間違いすら，もう一度入れ直しといったもので大変でした．

そのうち，ホーム・キーの[A]～[F]に左手を置き，右手でダンプ・リストを押さえながら[0]～[9]までの数字をテンキーで入れることに気づきましたが，やはり間違いが多くてデバッグに苦しみました．

そこで思いついたのが，この『Monitor-2』(以後，Ｍ２)です．このＭ２の利点は，

❶テンキーを16進キーとして使える．

❷ダンプ＆スクリーン・エディットができる．

❸チェック・サム機能が付いている．

などです．まずＭ２を走らせ，マシン語を片手で入力（キーインは倍速になります）してから，チェック・サムを調べ，スクリーン・エディタでデバッグをします．その後，BASICプログラムを入れるようにすれば，１人でも数時間でゲームが走るようになります．

M2の使い方

まず，リストを見てキーインしてください．そこでバグがいないことを確かめてください．そしてRUNします．すると

```
?WIDTH40, 25(1) or 80, 25(2)
```

と聞いてくるので，[1]か[2]を入力して[RETURN]キーを押してください．また，[RET]だけでは80，25に設定されます．次に，

```
SET(1) or DISPLAY(2) or CHECK SUM
(3)
```

と聞いてくるので，[1]，[2]，[3]のいずれかを入力してください．

■SET MEMORYの場合

まず，START＝と聞いてくるので，16進数でアドレスを入れてください．このとき，後の４桁が有効となります．これは，他のモードでも同じです．このときは[RET]キーが必要です．この状態ではテンキーはまだ16進キーとして使えません．

図１　SETMEM時の16進キー（▒部は無効）

HOME CLS	↑↓	→←	DEL INS
7	8	9	A
4	5	6	B
1	2	3	C
0	F	E	D

図２　[スペース]キーでデリートしたところ

```
D000:␣00=01␣00=23␣00=␣␣␣00=␣␣
D004:␣00=␣␣
```

図３　キー入力後にデータの誤まりを見つけたとき

```
                              ↗α
D000:␣00=01␣00=01␣00=(07)␣00=AB
D004:␣00=02␣00=21␣00=CF␣00=00
D008:␣00=03␣00=22␣00=FF␣00=07
D00C:␣00=_
          ↖いま，キーインしているところ
```

すると，画面が１度クリアされてから左上にアドレスが表示されて，後はＰＣのモニタと同じ要領です．そして，このSETMEMのときに限りテンキーは16進キーと変わります(図１)．

プログラムの上ではINKEYでＣＨＲを判別しているので，ホーム・キーの[A]～[F]は使えず，同じキャラクタはテンキーの場合と同じ現象を起します．

また，オート・リピートはきかないので，同じ文字が続くときは[R]キーを押してください．[R]を押す前に押したキーのオート・リピートとなります（ウァーApple II みたいだぞ～）．デリートは[スペース]キーです（図２）．

このとき画面の左上まできたときは，画面はダウン・スクロールされ，１番上の行にその前の行が現われます．また，アドレスを進めたいときは[S]キーを押すと進みます．このとき，アドレスの内容は元の状態です．

[スペース]や[S]は共に[R]キーでリピートがききます．ま

た，Zキーはどの場合も共通で，

SET(1)or DISPLAY(2)or CHECK SUM(3)

の初期状態になります．

また，図3のようなとき，αの場所にバグを見つけたとします．そのときはCキーを押してください．すると，そのアドレスが記憶されます．

スペースキーを押して，次にRキーを押しながら，αの場所へカーソル(?)を持って行き，書き直してからSキーを押します．

ここで，Rキーを押していると，Cキーを押したところでカーソルが止まり，〝ピー・ピー〞と音が出るので，またキーインを続けてください．

■DISPLAY MEMORYの場合

まず，START＝と表示し，次にEND＝と表示してくるので，それぞれ16進数で入力してください（やはり後の4桁が有効）．ENDアドレスを入力し終わってRETキーを押すと，アドレスが表示されて行きます（BASICのため，少し遅いのですが……）．

このとき，スペースキーを押せばカーソルは止まります．そして直したいところがあれば，↕，⇄キーでカーソルをコントロールして直したいところを直した後にRETキーを押してください．LINEINPUT文で行なっていますから，改行ごとにRETキーを押してください．

そのとき，次のことに注意してください．

❶アドレスを変えることはできます．

❷もともとスペースのあるところは，何を書いても関係ありません．

❸もともと数（0～F）のところに，スペースやその他の文字があると0とみなされます．

❹スクリーン・エディタから抜け出すときは，1度RETを押してください．その後でSキーを押すと，またダンプが続いて行きます（カーソルを止めたところから，キャリッジ・リターンをしないようエディットしてください）．また，その行が全部出てから，止めてください．

以上，慣れれば簡単なものです．この場合もZキーで初期状態になります．終わりまで行くと，EDITER?(y/n)と聞いてくるので，yを押せばエディタ・モード，nを押せば初期状態になります．

■CHECK SUMの場合

これはまず，START＝，END＝，BYTE＝と聞いてくるので，前者の2つは16進数で，BYTEは10進数で入力してください．すると，チェック・サムが始まります．

スペースキーはESCキーの代わりとなります．ここでも，Zキーで初期状態に戻ります．

終わりに一言

書き忘れましたが，SETのときもDISPLAYのときも関係ないキーを押したときは，音が出ます．

プログラムは，ダウン・スクロール以外はすべてBASICで書かれていて，これといったテクニックも使っていませんが，なかなか役に立つものです．

ぜひ使ってみてください．なお，テープ関係はモニタの方で行なってください．

写真1　セット・メモリ・コマンドの例
（途中でRキーを使用）

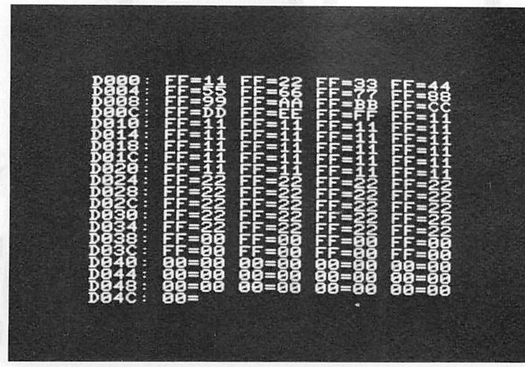


写真2　ディスプレイ・メモリ・コマンドの例
（エディタ・モードでカーソルを動かしているところ）

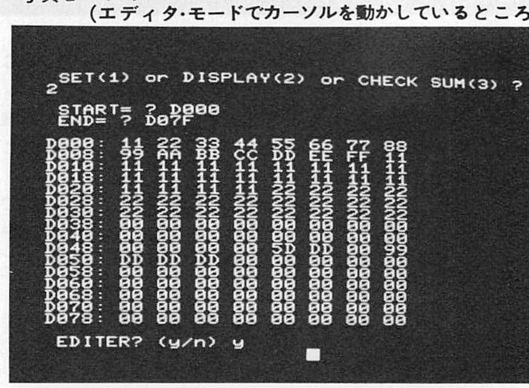


写真3　写真2の一部を修正する

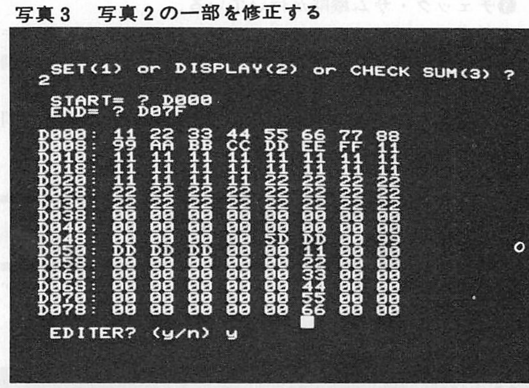
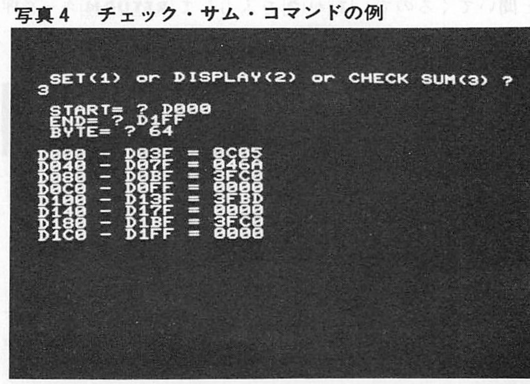


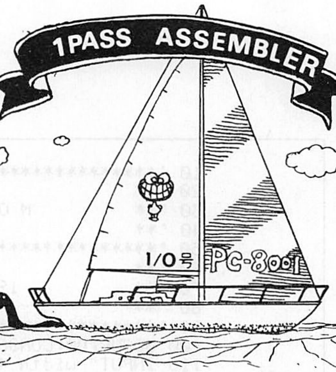
写真4　チェック・サム・コマンドの例



Monitor-2 プログラム・リスト

```
10 '**************************************************
20 '**                                              **
30 '**          M O N I T O R - P c   2             **
40 '**                                              **
50 '**************************************************
60 '**                                              **
70 '**      1980年10月23日  by-Koichi.Nakamura      **
80 '**                                              **
90 '**************************************************
100 GOSUB700:CONSOLE0,25,0,0:PRINTCHR$(12):DEFINTA-Z:DEFUSR0=&H8DA0
110 INPUT" width 40,25(1) or 80,25(2) ";W
120 IFW=1THENWIDTH40,25:GOTO140
130 W=2:WIDTH80,25
140 PRINTCHR$(12):LOCATE0,0:INPUT" SET(1) or DISPLAY(2) or CHECK SUM(3) ";A
150 IFA=2THEN400ELSEIFA=3THEN600ELSEIFA<>1THEN140
160 '****************   SET  MEMORY   ****************
170 PRINT
180 INPUT" START= ";S$:S$=RIGHT$(S$,4)
190 S=VAL("&h"+S$):D=0
200 PRINTCHR$(12):LOCATE0,0:PRINTRIGHT$("000"+HEX$(S),4);": ";:GOTO240
210 S=S+1:Z$=RIGHT$(HEX$(S),1):IFW=1THEN230
220 IFZ$="0"ORZ$="8"THENPRINT:PRINTRIGHT$("000"+HEX$(S),4);": ";:GOTO240ELSE240
230 IFZ$="0"ORZ$="4"ORZ$="8"ORZ$="C"THENPRINT:PRINTRIGHT$("000"+HEX$(S),4);": ";
240 PRINTRIGHT$("0"+HEX$(PEEK(S)),2);"=";:J$="":GOTO260
250 PRINTI$;:IFJ$=""THENJ$=I$ELSEJ$=J$+I$:PRINT" ";:POKES,VAL("&h"+J$):GOTO210
260 IFINP(4)=251THEN280
270 E$=INKEY$:IFE$=""THEN270ELSEIFE$="r"THEN280ELSEI$=E$
280 IFI$>"/"ANDI$<":"THEN250
290 IFI$=","THENI$="F":GOTO250
300 IFI$="."THENI$="E":GOTO250
310 IFI$=CHR$(13)THENI$="D":GOTO250
320 IFI$="="THENI$="C":GOTO250
330 IFI$="+"THENI$="B":GOTO250
340 IFI$="*"THENI$="A":GOTO250
350 IFI$<>" "THEN370ELSEIFPOS(0)=9ANDCSRLIN=0THENAA=USR0(0):I=S-4*W:LOCATE0,0:PR
INTRIGHT$("000"+HEX$(I),4);": ";:FORS=ITOI+4*W-1:L$=RIGHT$("0"+HEX$(PEEK(S)),2):
PRINTL$;"=";L$;" ";:NEXTS:S=S-1:PRINTSTRING$(3,29);"  ";STRING$(2,29);:J$="":GOT
O260
360 IFPOS(0)=9THENS=S-1:LOCATE24*W+3,CSRLIN-1:PRINT"  ";STRING$(2,29);:GOTO260EL
SEIF(POS(0)MOD2)=0THENPRINTCHR$(29);" ";CHR$(29);:J$="":GOTO260ELSES=S-1:PRINTST
RING$(6,29);"  ";STRING$(2,29);:GOTO260
370 IFI$<>"s"THEN390ELSEIF(POS(0)MOD2)=0THENPRINTCHR$(29);:PRINTRIGHT$("0"+HEX$(
PEEK(S),2);" ";ELSEPRINTRIGHT$("0"+HEX$(PEEK(S)),2);"";
380 IFI$="s"ANDS=V-1THENI$="¥":GOTO210ELSE210
390 IFI$="z"THENPRINT:GOTO140ELSEIFI$="c"THENV=S:GOTO260ELSEBEEP:GOTO260
400 '****************   DISPLAY  MEMORY   ****************
410 PRINT:INPUT" START= ";S$:S$=RIGHT$(S$,4)
420 INPUT" END= ";E$:E$=RIGHT$(E$,4)
430 S=VAL("&H"+S$):E=VAL("&h"+E$):PRINT:PRINTRIGHT$("000"+HEX$(S),4);": ";:GOTO4
70
440 Z$=RIGHT$(HEX$(S),1):IFW=1THEN460
450 IFZ$="0"THENPRINT:PRINTRIGHT$("000"+HEX$(S),4);": ";:GOTO470ELSE470
460 IFZ$="0"ORZ$="8"THENPRINT:PRINTRIGHT$("000"+HEX$(S),4);": ";
470 PRINTRIGHT$("0"+HEX$(PEEK(S)),2);" ";:IFS=ETHEN500ELSEI$=INKEY$:IFI$=""THENS
=S+1:GOTO440
480 IFI$=" "THENK=1:GOTO520
490 IFI$="z"THEN140ELSEBEEP:S=S+1:GOTO440
500 '*****************  EDITER MEMORY  *****************
510 PRINT:PRINT:PRINT" EDITER? (y/n) ";:IFINPUT$(1)="y"THEN530ELSEPRINT:GOTO140
520 X=POS(0):Y=CSRLIN:Q=S+1
530 IFINP(8)=253ORINP(8)=251THEN550
540 I$=INKEY$:IFI$=""THEN530ELSE560
550 LINEINPUTA$:S=VAL("&h"+LEFT$(A$,4)):FORI=7TO24*W+6STEP3:POKES,VAL("&h"+MID$(
A$,I,2)):S=S+1:NEXT:GOTO530
560 IFI$=" "ORI$=CHR$(13)OR(I$>CHR$(27)ANDI$<CHR$(32))THEN550
570 IFI$="s"ANDK=1THENLOCATEX,Y:S=Q:GOTO440
580 IFI$="z"THENPRINT:GOTO140
590 BEEP:GOTO530
600 '*****************    CHECK    SUM    *****************
610 PRINT:INPUT" START= ";S$:S$=RIGHT$(S$,4)
620 INPUT" END= ";E$:E$=RIGHT$(E$,4)
630 INPUT" BYTE= ";B
640 PRINT
650 S=VAL("&h"+S$):E=VAL("&h"+E$)
660 FORI=STOESTEPB:PRINTRIGHT$("000"+HEX$(I),4);" - ";RIGHT$("000"+HEX$(I+B-1),4
);" = ";
670 A=0:FORJ=0TO(B-1):A=A+PEEK(I+J):IFINKEY$<>""THENGOSUB690:NEXTJELSENEXTJ:PRIN
TRIGHT$("000"+HEX$(A),4):NEXTI
680 PRINT:GOTO140
690 IFINKEY$="z"THENPRINT:GOTO140ELSEIFINKEY$<>""THENRETURNELSEGOTO690
700 '*****************  MACHINE DATA !!!!  **************
710 RESTORE720:FORI=&H8DA0TO&H8DAB:READA$:POKEI,VAL("&H"+A$):NEXT:RETURN
720 DATA11,B7,FE,21,3F,FE,01,40,0B,ED,B8,C9
```

PC-8001用



PC-ASM

1パス セルフ・アセンブラ/逆アセンブラ

河村 秀明

PC-8001の外部に計測器を接続してインターフェイスを作ったり，BASIC プログラムをもっと高速で走らせようとすると，どうしても機械語でプログラムを組まなければなりません．一応，機械語モニタが本体に内蔵されていますが，ごく小さいプログラムならまだしも，多少プログラムが大きくなるといちいちハンド・アセンブルするにも限界があります．

現在，ROM やフロッピーディスク・ベースで PC-8001用のアセンブラが出回っているようです．わざわざ高い金をかけなくてもアセンブラ程度ならプログラムを組んでRAM 上で動かせるだろうと，アセンブラ・プログラムを作り始めました．

約1ヵ月かかって完全とまではいかないまでも，小さいプログラム（約1.5K Byte）なら一応実用になります．1パス セルフ・アセンブラ＋逆アセンブラ（これはずっと前に作ったもの）プログラム（略して PC-ASM）を完成したので発表します．

概　要

この PC-8001 用アセンブラ・プログラム（以下PC-ASM と略す）は，正確には1パス セルフ・アセンブラとラベル付加機能を持つ逆アセンブラとで構成されています．

アセンブラは，Zilog 社のZ80ニモニックで書かれたソース・プログラムをキーボードから入力すると，それに対応した機械語をダイレクトにメモリに格納します．また，逆アセンブラはメモリ内に格納されている機械語をZ80のニモニックで CRT あるいはプリンタ上に表示します．

このプログラムは N-BASIC を使って書かれていて，16K/32K 両システムで実行できるよう設計しました．プログラムの大きさは約9Kバイトで，16K システムではRAM領域のC000H～E3FFH番地までを占有します．PC-8001のモニタのワーキング・エリアがEA00Hから始まるので，このアセンブラによって機械語に変換したプログラムを格納できるメモリ領域はE400H～E9FFHまでの約1.5Kバイトです．

16K システムでも実行できるようにしたため，メモリの制約からZ80の全命令をアセンブル，または逆アセンブルすることができませんでした．アセンブルできる命令を機械語とニモニックに対応させて表1，表2に示します．また，逆アセンブルのみできる命令を同様に表3に示します．

表からもわかるように，このプログラムで処理できる範囲は8080の持つ全命令（ニモニックはZ80用に書く必要があります）+BIT，SET，RES，などの命令です．

インデックス・レジスタを使う命令はアセンブラ，逆アセンブラ共にできません．多少不備かもしれませんが，16Kシステムでは全命令のアセンブル，逆アセンブルをBASIC で実行するのは無理と思われます．

しかし，まだ多少メモリの余裕があるので，ユーザーのワークエリアを減らさずに逆アセンブラだけでも全命令実行させることは可能だと思います．6章でプログラムの説明をしています．余力のある方は改良してみてください．

アセンブラの仕様

1 ソース・プログラムのフォーマット

ソース・プログラムはラベル・フィールド，オペレーション・フィールド，オペランド・フィールドで構成されます（図1）．それぞれのフィールドの区切りは“␣”（スペース）で行ないます．

また，オペランド・フィールドは命令によって第1オペランド～第8オペランドまで記述でき，それぞれのオペランドの区切りは『，』（コンマ）で行ないます．

2 ソース・プログラムの入力方法

❶ アセンブラのソース・プログラムはキーボードから入力しますが，アルファベットの入力は大文字でも小文字でも，内部では大文字に変換され処理されます．

ただし，DC 命令（擬似命令の章で述べます）のオペランド・フィールドの文字列は大文字，小文字の区別が行なわれます．

以降の説明で共通となる，ソース・プログラムの入力フォーマットにおける記号の規則を表4にまとめます．

❷ Z80ソース・プログラムの入力フォーマットは，

〔ラベル〕␣ 〈命令のニモニック〉␣ 〔第1オペランド〕〔 ，第2オペランド〕 CR

表1　ニモニック—機械語対応表

下位＼上位	0	1	2	3	4	5	6	7
0	NOP	DJNZ e	JR NZ, e	JR NC, e	LD B, B	LD D, B	LD H, B	LD (HL), B
1	LD BC, nn	LD DE, nn	LD HL, nn	LD SP, nn	LD B, C	LD D, C	LD H, C	LD (HL), C
2	LD (BC), A	LD (DE), A	LD (nn), HL	LD (nn), A	LD B, D	LD D, D	LD H, D	LD (HL), D
3	INC BC	INC DE	INC HL	INC SP	LD B, E	LD D, E	LD H, E	LD (HL), E
4	INC B	INC D	INC H	INC (HL)	LD B, H	LD D, H	LD H, H	LD (HL), H
5	DEC B	DEC D	DEC H	DEC (HL)	LD B, L	LD D, L	LD H, L	LD (HL), L
6	LDB, n	LD D, n	LD H, n	LD (HL), n	LD B, (HL)	LD D, (HL)	LD H, (HL)	HALT
7	RLCA	RLA	DAA	SCF	LD B, A	LD D, A	LD H, A	LD (HL), A
8	EX AF, AF'	JR e	JR Z, e	JR C, e	LD C,B	LD E, B	LD L, B	LD A, B
9	ADD HL, BC	ADD HL, DE	ADD HL, HL	ADD HL, SP	LD C, C	LD E, C	LD L, C	LD A, C
A	LD A, (BC)	LD A, (DE)	LD HL, (nn)	LD A, (nn)	LD C, D	LD E, D	LD L, D	LD A, D
B	DEC BC	DEC DE	DEC HL	DEC SP	LD C, E	LD E, E	LD L, E	LD A, E
C	INC C	INC E	INC L	INC A	LD C, H	LD E, H	LD L, H	LD A, H
D	DEC C	DEC E	DEC L	DEC A	LD C, L	LD E, L	LD L, L	LD A, L
E	LD C, n	LD E, n	LD L, n	LD A, n	LD C, (HL)	LD E, (HL)	LD L, (HL)	LD A, (HL)
F	RRC A	RRA	CPL	CCF	LD C, A	LD E, A	LD L, A	LD A, A

下位＼上位	8	9	A	B	C	D	E	F
0	ADD A, B	SUB B	AND B	OR B	RET NZ	RET NC	RET PO	RET P
1	ADD A, C	SUB C	AND C	OR C	POP BC	POP DE	POP HL	POP AF
2	ADD A, D	SUB D	AND D	OR D	JP NZ, nn	JP NC, nn	JP PC, nn	JP P, nn
3	ADD A, E	SUB E	AND E	OR E	JP nn	OUT (n), A	EX (SP), HL	DI
4	ADD A, H	SUB H	AND H	OR H	CALL NZ, nn	CALL Nc ,nn	CALL PO,nn	CALL P, nn
5	ADD A, L	SUB L	AND L	OR L	PUSH BC	PUSH DE	PUSH HL	PUSH AF
6	ADD A, (HL)	SUB (HL)	AND (HL)	OR (HL)	ADD A, n	SUB n	AND n	OR n
7	ADD A, A	SUB A	AND A	OR A	RST 0	RST 10H	RST 20H	RST 30H
8	ADC A, B	SBC A, B	XOR B	CP B	RET Z	RET C	RET PE	RET M
9	ADC A, C	SBC A, C	XOR C	CP C	RET	EXX	JP (HL)	LD SP, HL
A	ADC A, D	SBC A, D	XOR D	CP D	JPZ, nn	JP C, nn	JP PE,nn	JP M, nn
B	ADC A, E	SBC A, E	XOR E	CP E	表2参照	IN A, (n)	EX DE, HL	EI
C	ADC A, H	SBC A, H	XOR H	CP H	CALL Z, nn	CALL C, nn	CALL PE, nn	CALL M, nn
D	ADC A, L	SBC A, L	XOR L	CP L	CALL nn		表3参照	
E	ADC A, (HL)	SBC A, (HL)	XOR (HL)	CP (HL)	ADC A, n	SBU A, n	XOR n	CP n
F	ADC A, A	SBC A, A	XOR A	CP A	RST 8	RST 18H	RST 28H	RST 38H

●［薄い網］は2バイト命令［濃い網］は3バイト命令　●オペランドのnは1バイトの数値　nnは2バイトの数値
eは−128〜＋127の2の補数



です．

ラベルは省略できますが，ラベルの有無にかかわらず，『␣』を必ず1個入れラベル・フィールドの終了とします．

命令は省略できないのでニモニック（表1〜表2）で入力し，また『␣』を1個入れ，オペレーション・フィールドの終了とします．

オペランドは命令によってはないものもあるので，その場合は省略できますが，持つものについては第1オペランドあるいは第2オペランド（区切りは『，』で行ないます）を入力します．

ソース・プログラムの終了は **CR** で，このキーが入力されるまではアセンブルは実行されません．

3 ラベル・フィールド

ラベルに関する規則は，次のようになっています．

❶　ラベルは先頭が英大文字A〜Zで始まる6文字以内の文字列です．先頭以外の文字には数字0〜9およびASCII記号（『␣』（スペース）および『，』および『＇』は使用不可）およびグラフィック記号も使用できます．

ただし，グラフィックはプリンタによっては印字されないものがあります．

```
例）HEX12   EQU  12H
    INP$    EQU  1754H
    OUT♥    EQU  0E500H
```

❷　レジスタ名はラベルに使用できません．以下のレジスタをラベルとして使用した場合，ラベル・フィールドにおいては登録の際にエラーは生じませんがオペランド・フィールドにおいては以下の例のように命令によってレジスタと解釈される場合があるので使用できません（5オペランド・フィールド参照）．

レジスタ	A,B,C,D,E,F,H,L,I,R, AF,BC,DE,HL,SP,IX,IY

表2　ニモニック—機械語対応表

下位＼上位	0	1	2	3	4	5	6	7
0	RLC B	RL B	SLA B		BIT 0, B	BIT 2, B	BIT 4, B	BIT 6, B
1	RLC C	RL C	SLA C		BIT 0, C	BIT 2, C	BIT 4, B	BIT 6, C
2	RLC D	RL D	SLA D		BIT 0, D	BIT 2, D	BIT 4, D	BIT 6, D
3	RLC E	RL E	SLA E		BIT 0, E	BIT 2, E	BIT 4, E	BIT 6, E
4	RLC H	RL H	SLA H		BIT 0, H	BIT 2, H	BIT 4, H	BIT 6, H
5	RLC L	RL L	SLA L		BIT 0, L	BIT 2, L	BIT 4, L	BIT 6, L
6	RLC (HL)	RL (HL)	SLA (HL)		BIT 0, (HL)	BIT 2, (HL)	BIT 4, (HL)	BIT 6, (HL)
7	RLC A	RL A	SLA A		BIT 0, A	BIT 2, A	BIT 4, A	BIT 6, A
8	RRC B	RR B	SRA B	SRL B	BIT 1, B	BIT 3, B	BIT 5, B	BIT 7, B
9	RRC C	RR C	SRA C	SRL C	BIT 1, C	BIT 3, C	BIT 5, C	BIT 7, C
A	RRC D	RR D	SRA D	SRL D	BIT 1, D	BIT 3, D	BIT 5, D	BIT 7, D
B	RRC E	RR E	SRA E	SRL E	BIT 1, E	BIT 3, E	BIT 5, E	BIT 7, E
C	RRC H	RR H	SRA H	SRL H	BIT 1, H	BIT 3, H	BIT 5, H	BIT 7, H
D	RRC L	RR L	SRA L	SRL L	BIT 1, L	BIT 3, L	BIT 5, L	BIT 7, L
E	RRC (HL)	RR (HL)	SRA (HL)	SRL (HL)	BIT 1, (HL)	BIT 3, (HL)	BIT 5, (HL)	BIT 7, (HL)
F	RRC A	RR A	SRA A	SRL A	BIT 1, A	BIT 3, A	BIT 5, A	BIT 7, A

下位＼上位	8	9	A	B	C	D	E	F
0	RES 0, B	RES 2, B	RES 4, B	RES 6, B	SET 0, B	SET 2, B	SET 4, B	SET 6, B
1	RES 0, C	RES 2, C	RES 4, C	RES 6, C	SET 0, C	SET 2, C	SET 4, C	SET 6, C
2	RES 0, D	RES 2, D	RES 4, D	RES 6, D	SET 0, D	SET 2, D	SET 4, D	SET 6, D
3	RES 0, E	RES 2, E	RES 4, E	RES 6, E	SET 0, E	SET 2, E	SET 4, E	SET 6, E
4	RES 0, H	RES 2, H	RES 4, H	RES 6, H	SET 0, H	SET 2, H	SET 4, H	SET 6, H
5	RES 0, L	RES 2, L	RES 4, L	RES 6, L	SET 0, L	SET 2, L	SET 4, L	SET 6, L
6	RES 0, (HL)	RES 2, (HL)	RES 4, (HL)	RES 6, (HL)	SET 0, (HL)	SET 2, (HL)	SET 4, (HL)	SET 6, (HL)
7	RES 0, A	RES 2, A	RES 4, A	RES 6, A	SET 0, A	SET 2, A	SET 4, A	SET 6, A
8	RES 1, B	RES 3, B	RES 5, B	RES 7, B	SET 1, B	SET 3, B	SET 5, B	SET 7, B
9	RES 1, C	RES 3, C	RES 5, C	RES 7, C	SET 1, C	SET 3, C	SET 5, C	SET 7, C
A	RES 1, D	RES 3, D	RES 5, D	RES 7, D	SET 1, D	SET 3, D	SET 5, D	SET 7, D
B	RES 1, E	RES 3, E	RES 5, E	RES 7, E	SET 1, E	SET 3, E	SET 5, E	SET 7, E
C	RES 1, H	RES 3, H	RES 5, H	RES 7, H	SET 1, H	SET 3, H	SET 5, H	SET 7, H
D	RES 1, L	RES 3, L	RES 5, L	RES 7, L	SET 1, L	SET 3, L	SET 5, L	SET 7, L
E	RES 1, (HL)	RES 3, (HL)	RES 5, (HL)	RES 7, (HL)	SET 1, (HL)	SET 3, (HL)	SET 5, (HL)	SET 7, (HL)
F	RES 1, A	RES 3, A	RES 5, A	RES 7, A	SET 1, A	SET 3, A	SET 5, A	SET 7, A

●空白は無効命令　●この表は先頭がＣＢで始まる命令群

```
例)  B   EQU  10       ←Bというラベルに10を与える
     BB  EQU  20       ←BBというラベルに20を与える
     ␣LD␣A,BB          ←この場合は Aレジスタ←20
                         という命令だが,
     ␣LD␣A,B           ←この場合はAレジスタ←B
                         レジスタという命令になる
```

❸　登録できるラベル数は，未知のラベル（5オペランド・フィールドを参照）と合わせて31です．これを超えてラベルを使用すると"LBL-MAX"が印字され，それ以後のラベルの登録はできません．もちろん，既に登録されたラベルはオペランド・フィールドに使用することができます．

ラベル数31は少ないと感じるかもしれませんが，BASICで実行しているため，ラベルが増えるとそれに比例してラベル処理に時間がかかり，せいぜいふやして実用になるのはラベル数が50個程度まででです．

4 オペレーション・フィールド

オペレーション・フィールドにはＺ80の命令あるいは後述の擬似命令のニモニックを記入します．Ｚ80のニモニックはZilog 社のタイプのもので，この PC-ASM でアセンブルできる命令を表１，表２に示します．

この表にない命令（インデックス・レジスタを使う命令や表３の命令）を使用する場合はハンド・アセンブルした後その機械語を後述のDB命令で入力してやってください．

```
例) ␣LD␣IX,1234H ←入力したいソース・プ
                   ログラム
  ⇒ ␣DB␣0DDH,21H,1234H  ←DB命令で
                           入力する.
ハンド・
アセンブル
```

表3　ニモニック―機械語対応表

下位＼上位	0	1	2	3	4	5	6	7
0					IN B, (C)	IN D, (C)	IN H, (C)	
1					OUT (C), B	OUT (C), D	OUT (C), H	
2					SBC HL, BC	SBC HL, DE	SBC HL, HL	SBC HL, SP
3					LD (nn), BC	LD (nn), DE		LD (nn), SP
4					NEG			
5					RET N			
6					IM 0	IM 1		
7					LD I, A	LD A, I	RRD	
8					IN C, (C)	IN E. (C)	IN L, (C)	IN A, (C)
9					OUT (C), C	OUT (C), E	OUT (C), L	OUT (C), A
A					ADC HL, BC	ADC HL, DE	ADC HL HL	ADC HL, SP
B					LD BC, (nn)	LD DE, (nn)		LD SP, (nn)
C								
D					RET I			
E						IM 2		
F							RLD	

下位＼上位	8	9	A	B	C	D	E	F
0			LD I	LD IR				
1			CP I	CP IR				
2			IN I	IN IR				
3			OUT I	OT IR				
4								
5								
6								
7								
8			LDD	LDD R				
9			CPD	CPD R				
A			IN D	IND R				
B			OUT D	OTDR				
C								
D								
E								
F								

●この表は先頭がEDで始まる命令群　●空白は無効命令
●逆アセンブラのみできる命令　●（網掛け）は4バイト命令

5 オペランド・フィールド

オペランド・フィールドは『，』で区切られたオペランドで構成されていて最大第8オペランドまで記入できます．Z80の命令で使用するオペランドは第2オペランドまでで，第3オペランド以上を記入できるのは後述のDB 命令だけです．

各オペランドには命令に対応したレジスタ名，定数，アドレス，相対ジャンプ命令のオフセット値などを記入します．定数やアドレスやオフセット値の代わりにラベルを使用することができます．

なお，オペランドの処理で解釈される順序は

1）レジスタ群（B，C，D，E，H，L，(HL)，A，命令によってBC，DE，HL，SPなど）
2）条件群（NZ，Z，NC，C，PO，PE，P，M）
3）定数　①10進
　　　　②16進
4）ラベル

となっています．



❶ 使用できる定数について

a）10進数…10進数（0～9）で構成された数です．範囲は0≦n≦65535までです．

例）　␣LD␣BC,$12345

b）16進数…16進数（0～9とA～F）で構成された数です．ただし，16進数の始めが文字（A～F）の場合はラベルと区別するために必ず0を付けます．数値の範囲は0H～0FFFFHです．

例）　CR␣EQU␣0DH LD␣BC, 12FFH

❷ 使用できるラベルについて

a）定数の代用としてのラベル

ラベルは定数の代わりにオペランド・フィールドに記入

表4　入力フォーマットの規則

	記 号	説　　明
I	□	キーボードから入力する文字 例）a…小文字のa A…大文字のA ␣…スペース CR…キャリッジ・リターン
II	[　　]	省略できるフィールドを示す．
III	<　　>	省略できないフィールドを示す．

図1　ソース・プログラムのフィールドの種類

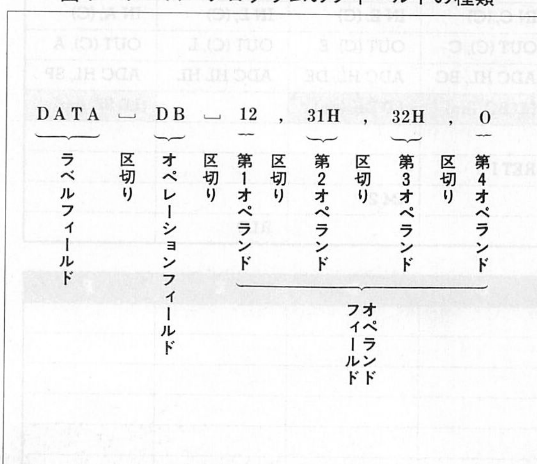
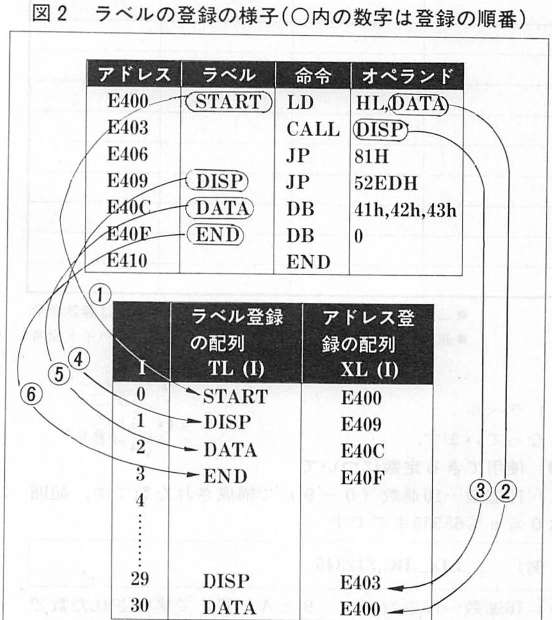


図2　ラベルの登録の様子(○内の数字は登録の順番)



することができますが，その場合そのラベルは既に定義されていなければなりません．

b）アドレスの代用としてのラベル

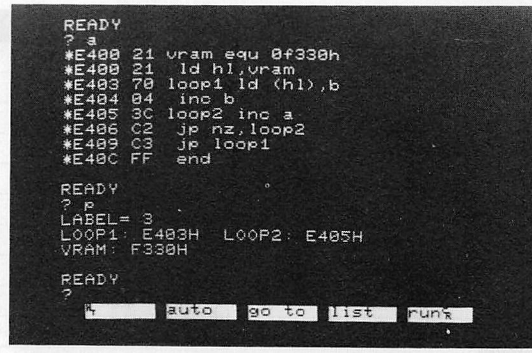
アドレスの代わりにラベルを使用した場合，そのラベルが既に定義されているときはそのアドレスをラベル登録エリアから引き出し，オペランドの処理は終了します．

しかし，そのラベルが未定義ラベルの場合は，そのアドレスが与えられないので，未知ラベルとして，いったんラベル登録エリアの最後部から順に，ラベル名およびそのラベルが使われた命令の先頭アドレスを登録します（図2）．

このように未知ラベルはラベル登録エリアを見かけ上1個分減少させるのでラベル登録エリアをオーバーしないように注意が必要です．

ここで登録された未知ラベルにアドレスを与えるのは後述の **END** 命令なので，この命令の前にそのラベルを定義しなければなりません．

写真1　アセンブル終了後，ラベル・テーブルを確認する



擬似命令

擬似命令は以下の6種類があります．これらの命令はアセンブラのときのみ用いることができます．この擬似命令により作成されたオブジェクト・コードは，逆アセンブルするとそれに対応したＯＰコードのニモニックでリストに出力されます．

1 ORG（origin）命令

❶**ORG** 命令は，アセンブラの開始アドレスを定義，あるいは変更する命令で，オペランドの値がそのアドレスを与えます．

❷オペランドの値は，16K システムではＥ４００Ｈ～Ｅ９ＦＦＨまで，32K システムではＡ４００Ｈ～Ｅ９ＦＦＨまでの範囲です．この範囲を越えてオブジェクト・プログラムをストアするとコンピュータが暴走する場合があります．

❸ **ORG** 命令の入力フォーマットは，

〔ラベル〕 ␣ORG␣ 〈絶対番地(10進または16進)またはラベル〉

で，オペランド・フィールドのラベルはそれ以前に定義されていなければなりません（図3）．

❹ **RUN** 直後のアセンブラ開始アドレスは，**ORG** 命令で定義しなくても16K システムではＥ４００Ｈ，32K システムではＡ４００Ｈに自動的にセットされます．

2 EQU（equate）命令

❶ **EQU** 命令は，ラベル・フィールドのラベルの値を定義する命令です．

❷ **EQU** 命令の入力フォーマットは

〈ラベル〉 ␣EQU␣ 〈定数(10進または16進)〉

で，定数は０～65535，または０Ｈ～０ＦＦＦＦＨまでの値です（図４）．

❸ EQU 命令はオブジェクト・コードは作成されないので，ソース・プログラムの任意の位置で使うことができます．

3 DB (define byte) 命令

❶ DB 命令はオペランド・フィールドに並んでいる定数，またはラベルの値を１バイトまたは２バイトのデータとしてメモリに確保する命令です．

❷オペランドで使用できるラベルはそれ以前に定義されているものに限ります．

❸オペランド・フィールドには最大８個の定数，またはラベルを記述することができます．

❹オペランドの値が０～255 のときは１バイトのデータを与え，256～65535のときは２バイトのデータを与えます．ただし，２バイト・データの場合は，格納される第１データは数値の下位８ビットで第２データが上位８ビットとなるので注意してください．

❺ DB 命令の入力フォーマットは

〔ラベル〕␣DB␣〈定数(10進または16進)またはラベル [，定数またはラベル…]

となります（図５）．

4 DC (define characters) 命令

❶ DC 命令は，オペランド・フィールドの『▼』でくくられたASCII 文字，カタカナ，および PC-8001のグラフィック記号などをキャラクタ・コード（PC-8001リファレンス・マニュアルp.104）に変換し，メモリに確保する命令です．

❷記述できる文字例の長さは登録されているラベルの文字数と関連しますが（詳しくは次章のａ コマンドを参照），最低の場合でも約20個です．

❸ DC 命令の入力フォーマットは

〔ラベル〕␣DC␣▼〈任意の文字列〉▼

です．

5 DS (define strage) 命令

❶ DS 命令は，オペランドの値と同数のバイト分のメモリ領域を確保する命令です．メモリの内容は変化しません．この命令が実行されたあとのアセンブル開始アドレスはオペランドの値だけ前よりも増加します．

❷オペランドで使用できるラベルはそれ以前に定義されているものに限ります．

❸ DS 命令の入力フォーマットは

〔ラベル〕␣DS␣〈定数(10進または16進)またはラベル〉

です（図７）．

6 END (end of assembly) 命令

❶ END 命令はソース・プログラムの最後に必ずつけ，アセンブルを終了させます．

❷ END 命令の入力フォーマットは

␣END

です．

❸PC-ASM はこの END 命令を受けて次の処理をします．

図３　ＯＲＧ命令の実行例

```
*E400_FF__org_0E500h
*E500_FF_■
```

図４　ＥＱＵ命令の例

```
CLEAR_EQU_12
CR_EQU_0DH
LF_EQU_0AH
```

図５　ＤＢ命令の例

```
DATA_EQU_1234H
_DB_0, 10, 10h, DATA
```

アセンブル
されたデータ

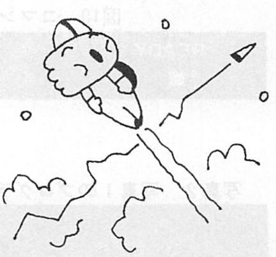
```
00  0A  10  34 12
```

図６　ＤＣ命令の例

```
TITLE_DC_▼***PC-ASM***▼
TIME_DC_▼10時30分54秒▼
NAME_DC_▼カワムラ▼
```

図７　ＤＳ命令の実行例

```
*E400_FF_AREA_DS_10H
*E410_FF_■
```



1）オペランド・フィールドで使われた未知ラベルのアドレスを登録ラベルの中から捜し出します．

2）もし，登録ラベル中に見つかったときは４）の処理へ進みます．

3）登録ラベル中に見つからなかったときは，そのラベルが用いられた命令のオペランドは正しい値ではないので，未定義ラベル（undefined label）があったことを

UNDF␣〈ラベル名〉␣〈そのラベルが使われた命令の先頭アドレス〉

の順にディスプレイに表示します．そして，１）へ戻ります．

4）相対ジャンプ命令のオペランドに使われた場合はそのオフセット値を，その他の場合はその値を示す2バイト・データを元のＯＰコードが格納されている部分にストアし，正しいオペランドになり，１）へ戻ります．

5）未知ラベルの処理が終わると“READY”を表示しコマンド待ちとなります．

図8　16Kシステムのメモリ・マップ

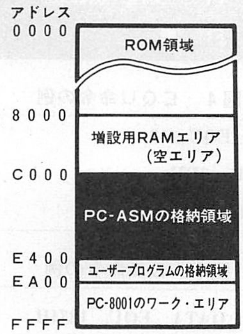


図9　32Kシステムのメモリ・マップ

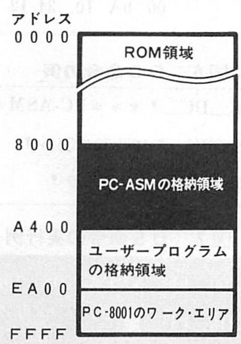
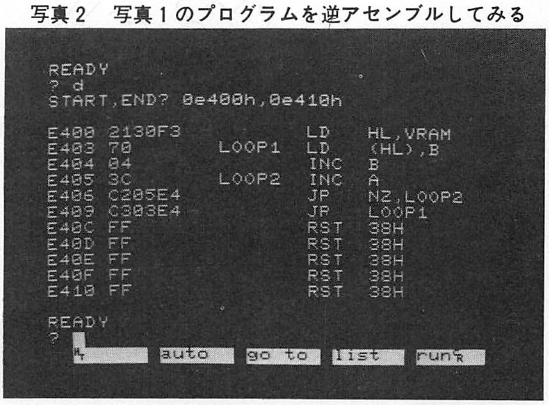


図10　コマンド待ちの状態

```
READY
?■
```

写真2　写真1のプログラムを逆アセンブルしてみる



❹未知ラベルはEND命令により，ラベル領域から削除されますが，登録ラベルは残っています．そこでdコマンドのアドレス入力の際にアドレスの代用としてラベルを使用することができます．

再びアセンブルを開始しても，同様にラベルは消えません．ラベルを消したいときは再びRUNさせてください．

PC-ASMの使用法

1 メモリ・マップ

PC-ASMプログラムは16Kシステム上で作りました．このプログラムのメモリ上での専有領域を図8に示します．32Kシステムでもプログラムの変更なしで動作しますが，表5のように変更すれば，より大きなプログラムのアセンブルが可能となります．変更後の32Kシステムでのメモリの専有領域を図9に示します．プログラムを**リスト1**に示します．

2 入力コマンド

PC-ASMをRUNさせると図10の状態となりコマンドの入力待ちとなります．PC-ASMのコマンドは以下の4種類あり，キーインは小文字で行ないます．

1）　a　(assembler) アセンブルを行なう．
2）　d　(dis assembler) 逆アセンブルを行なう．
3）　p　(print label) シンボル・リストを出力します．
4）　e　(end) PC-8001のモニタに戻る．

❶ a (assembler) コマンド

コマンド待ちの状態図10で **a** **CR** をキーボードから入力します．すると，図11に示すように＊に続いて，アセンブル後機械語を格納する先頭アドレスを16進数の4文字で表示し，そのあと␣（スペース）に続いてそのアドレスのデータを16進数の2文字で表示しライン入力状態となります．

RUN直後の格納開始アドレスは16KシステムではE400H，32KシステムではA400Hになります．

図11のカーソルのある位置が第1カラムとなります．入力できる最大カラム数は，登録されているラベル数と関連しています．正確には下の式で与えられますが，実際的にはソース・プログラムを1行以内（1行＝40文字として）におさえれば問題はありません．

> **入力できる最大カラム数≒（270－Σ（登録されているラベルの文字数））/2**

もし，5文字のラベルを30個登録しているような最低の場合だと約60となります．アセンブラのソース・プログラムの入力はライン入力となっているので **CR** キーを押すまでは訂正することができます．

CR キーが押されるとその行のソース・プログラムは機械語に変換され＊に続く格納開始アドレスから順にストアされていきます．

もし，ソース・プログラムを入力せずに **CR** キーだけを押すとそのアドレスのメモリの内容は変更されずに，格納開始アドレスが1つだけカウントupします．

❷ d (disassembler) コマンド

コマンド入力待ちの状態で，**d** **CR** をキーボードから入力すれば，図12に示すように逆アセンブルの開始アドレスと終了アドレスの入力待ちとなります．

スタート，エンド・アドレスは『，』で区切り，10進数，16進数またはラベルで入力します．

図13に逆アセンブラの実行例を示します．逆アセンブラにおけるカラムとフィールドは図14のような対応があります．

なお，逆アセンブラ・リストをプリンタに打ち出したいときは，PC-ASMプログラムの行番号3030の**PRINT**を

図11　aコマンドの実行後の状態

（▢内の文字はキーボードからの入力です
CRはキャリッジ・リターンです.）

```
READY
?a CR
*E400 ■
```

図12　dコマンドの実行後の状態

```
READY
?d CR
START, END ?■
```

図13　逆アセンブラの実行例

```
START, END ?begin,0e405h

E400␣3E0A␣␣␣␣␣BEGIN␣␣LD␣␣␣A,0AH
E402␣3210E4               LD␣␣␣(E410H),A
E404␣CD20E4     LOOP␣     CALL␣E420H

READY
?■
```

図15　pコマンドの実行例

```
READY
?p CR
LABEL＝4
BEGIN：E400H    LOOP：E404H
DATA ：E410H    END：E430H

READY
? ■
```

図14　カラムとフィールドの種類

カラム	1　～　4	5	6　～　13	14	15　～　20	21	22　～　25	26	27　～　40
例	E400	␣	3E0A␣␣␣␣	␣	BEGIN␣	␣	LD␣␣	␣	A, 0AH
フィールド	アドレス・フィールド	ブランク	オペコード・フィールド	ブランク	ラベル・フィールド	ブランク	オペレーション・フィールド	ブランク	オペランド・フィールド

表5　32Kシステムでの変更点

line No.	変更前	変更後
110	clear 350, & h E3FF：～	clear 350, & h A3FF：～
390	～：Z＝58368！：～	～：Z＝41984！：～

LPRINT に変更してください.

❸ p (print label) コマンド

コマンド入力待ちの状態で p CR をキーボードから入力した例を図15に示します.

このコマンドは現在登録されているラベルをアドレスの小さい順から表示します.

最初に表示される"LABEL＝N"（N＝0～31）は現在登録されているラベル数がn個あることを示します. なお, ラベル・リストをプリンタに打ち出したいときは, PC-ASMプログラムの行番号3430にある3個のPRINTをLPRINTに変更してください.

❹ e (end) コマンド

プログラムを終了し, PC-8001のBASICモニタに戻ります.



エラーメッセージ

エラーメッセージはアセンブラを実行時にエラーが発生したときにディスプレイへ出力されます.

以下に, エラーメッセージの種類とその内容を列挙します.

1 ER－のエラーメッセージ

以下の4つのエラーが発生したときはアセンブルが不能なので, そのアドレスで再度入力待ちになります.

❶ ER-LBL　(ERROR LABEL)

ラベルの記述に誤りがある.

a）ラベルの先頭に英大文字A～Z以外の文字が使われた.

b）ラベルが6文字を越えた.

❷ ER-INS　(ERROR INSTRUCTION)

命令のニモニックの記述に誤りがある.

a）Z80にない命令を使った(インデックス・レジスタを使う命令なども含まれる).

❸ ER-OP　(ERROR OPERAND)

オペランドの記述に誤りがある.

a）未定義ラベルを定数の代わりに使った.

b）定数の記述あるいはその値（0～65535 または0H～0FFFFH）に誤まりがある.

c）ラベルあるいはレジスタの記述に誤りがある.

d）2バイト命令におけるオペランドの値が0～255 の範囲を越えた.

❹ ER-OVR　(ERROR OVER)

相対ジャンプ命令 (DNJZ, JR 命令) において，そのオフセット値が－128～＋127の範囲を越えた.

2 LBL－ のエラーメッセージ

以下の２つのエラーはラベル登録時に生じたエラーで，アセンブラは続行されます.

❶ **LBL-MAX** (LABEL MAXIMUM)

『登録ラベル数＋未知ラベル数』が31個を越えた.

a）このエラーが生じた場合，そこで使われたラベルを含めそれ以後はラベルの登録並びに未知ラベルの使用が禁じられます.

b）この場合，それ以前に登録された未知ラベルは END 命令によって，アドレスが未定義となるので未定義ラベルとして一括してラベル名とそれが使われたアドレスが表示されます．詳しくは END 命令を参照.

❷ **LBL-DBL** (LABEL DOUBLE)

登録されているラベルを再定義した．そのラベルは登録されません.

PC-ASMプログラムの説明

プログラムの全体のフローチャートを図16に，行番号と処理ルーチンの説明を表６に，プログラムに使用されている主要な変数とその説明を表７に示します.

また，プログラミング例を実行例１に，逆アセンブルしたリストを実行例２に示します.

図16 フローチャート

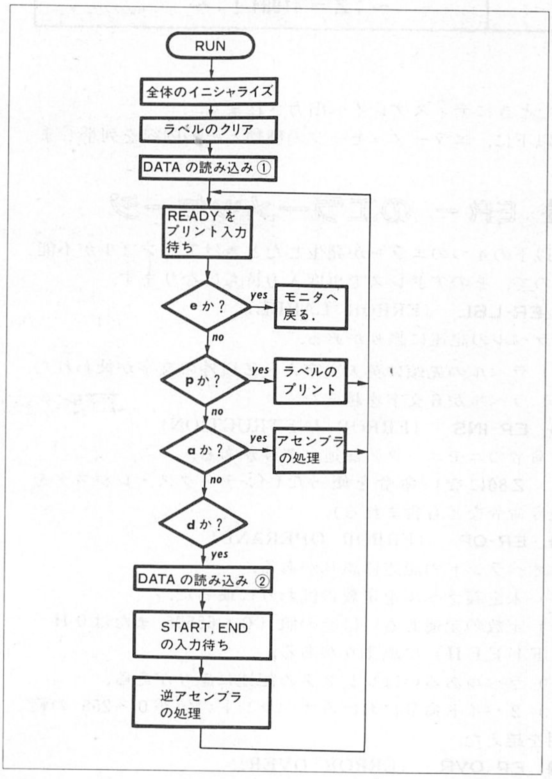


表６ プログラムの説明

行番号	説明
110	文字変数の格納領域およびBASICの格納領域の上限の制限，文字変数，整数変数の定義
120～250	アセンブラ，逆アセンブラで使用するデータ
260～350	逆アセンブラで使用するデータ
360～390	逆アセンブラで使用する機械語の文字群
400	キャラクタ→16ビット・データ変換
410	16ビットデータから下位８ビットを取り出す.
420	８ビットデータを２個の16進に変換，16ビットデータを４個の16進に変換
430	配列のディメンションを宣言
440	データの読み込み①
1000～1040	READY印字で入力コマンド待ち.
1050～3290	逆アセンブラのサブルーチン
1050～1080	データの読み込み②
1100	START, END のアドレス入力待ち
2000～2120	逆アセンブラの処理（規則性の大きい場合）
3000～3040	逆アセンブラ・リストの出力処理
3050～3150	逆アセンブラの処理（規則性のある場合）
3160～3290	CB，EDで始まる命令の逆アセンブル
3300～3360	ラベル登録サブルーチン
3380～3410	ラベルのアドレスを捜すサブルーチン
3420～3430	ラベル・プリント（p コマンド）を行なうサブルーチン
4000～6100	アセンブラ・サブルーチン
4010	各行ごとのイニシャライズ
4020	ソース・プログラムの入力待ち
4030～4070	ソース・プログラムのフィールドおよびオペランドの分離
4080～4200	擬似命令の処理
4100	ORG命令の処理
4110	EQU命令の処理
4120	DS命令の処理
4130～4170	END命令の処理
4180～4190	DB命令の処理
4200	DC命令の処理
4210～4230	１バイト命令の処理
4240～4300	"LD"命令の処理
4310～4350	"ADD"～"CP"までの命令の処理
4360～4390	規則性のない命令の処理
4400～4440	"JP", "CALL"命令の処理
4450～4460	"INC", "DEC"命令の処理
4470～4500	相対ジャンプ命令の処理
4510～4520	"POP", "PUSH"命令の処理
4530～4570	"EX", "OUT", "IN"命令の処理
5000～5050	アセンブルされたDATAの格納
6000～6030	10進数，16進数を数値へ変換する.
6040～6100	ラベルを数値あるいはアドレスに変換する.
6080	未知ラベルの登録を行なう.



表７ 主要な変数と説明

	変数名	説明
アセンブラ，逆アセンブラ共通	L	登録されているラベル数
	LL	登録されている未知ラベル数
	O(8)	命令のニモニック
	T(5,7)	オペランドに使用されるレジスタや条件など
	TL(30)	ラベルの格納領域
	XL(30)	アドレスの格納領域

逆アセンブラ	XX	逆アセンブル中の命令先頭アドレス
	X	現在逆アセンブルしているアドレス
	S	命令のニモニック
	TA,TB	オペランドの内容
	KK	X番地の内容（0～255）
アセンブラ	W (1)	ソース・プログラム中のラベル
	W (2)	ソース・プログラム中の命令
	W(3)～(10)	第1～第8オペランド
	Z	機械語の格納アドレス
	X	現在アセンブル中の先頭アドレス

	D	格納される命令の機械語
	D1,D2	格納されるオペランドの下位8ビット上位8ビット
	W	現在処理中の文字列（ニモニックやオペランドなど）
	XN	オペランドに使われた定数やラベルの値（16ビット）
	G	ソース・プログラムが何バイト命令かを示す.

実行例1　アセンブラの実行

```
READY
a
*E400 21 print equ 52edh
*E400 21 pc8001 equ 81h
*E400 21 loop ld hl,data
*E403 CD  call print
*E406 3A  ld a,(count)
*E409 3D  dec a
*E40A 32  ld (count),a
*E40D 20  jr nz,loop
*E40F C3  jp pc8001
*E412 4E data dc 'PC-8001 Assembler'
*E423 62  db 32,32,0
*E426 00 count db 10
*E427 20  end
```

実行例2　逆アセンブルしたソース・プログラム

```
READY
d
E400 2112E4    LOOP  LD    HL,DATA
E403 CDED52          CALL  PRINT
E406 3A26E4          LD    A,(COUNT)
E409 3D              DEC   A
E40A 3226E4          LD    (COUNT),A
E40D 20F1            JR    NZ,LOOP
E40F C38100          JP    PC8001
E412 50        DATA  LD    D,B
E413 43              LD    B,E
E414 2D              DEC   L
E415 3830            JR    C,E447H
E417 3031            JR    NC,E44AH
E419 2041            JR    NZ,E45CH
E41B 73              LD    (HL),E
E41C 73              LD    (HL),E
E41D 65              LD    H,L
E41E 6D              LD    L,L
E41F 62              LD    H,D
E420 6C              LD    L,H
E421 65              LD    H,L
E422 72              LD    (HL),D
E423 2020            JR    NZ,E445H
E425 00              NOP
E426 0A        COUNT LD    A,(BC)

READY
p
```

リスト1　PC-ASM プログラム

```
100 '*** PC-ASM May 21 1980 ***
110 CLEAR 350,&HE3FF:DEFINT A-N:DEFSTR O-W
120 O(1)="ADD ADC SUB SBC AND XOR OR  CP  "
130 O(2)="RLCARRCARLA RRA DAA CPL SCF CCF "
140 O(3)="LD  INC DEC RET JP  CALLRST"
150 O(4)="LD  INC DEC ADD POP PUSH"
160 O(5)="NOP RET EXX DI  EI  HALT"
170 O(6)="LD  EX  DJNZJR  JP  CALLIN  OUT"
180 O(7)="(BC)A(BC) (DE)A(DE)SPHL"
190 O(8)="RLC RRC RL  RR  SLA SRA     SRL BIT RES SET "
200 DATA ,BIT,RES,SET,LD,CP,IN,OUT
210 DATA B,C,D,E,H,L,(HL),A
220 DATA NZ,Z,NC,C,PO,PE,P,M
230 DATA 0,8,10H,18H,20H,28H,30H,38H
240 DATA BC,DE,HL,SP,BC,DE,HL,AF
250 DATA A,A,,A,,,,
260 DATA 0,8,0,8,&H11,&H12,&H21F0,&HC
270 DATA &H23,8,&H62F4,0,&H62F5,8,&H25FB,&H36
280 DATA 0,&H64F0,&H41,&H1043,&H42,&H44,&H45,0
290 DATA &H180,&H671,&H810,&H40F2,&H380,&H40F3,&H830,&H4BF3
300 DATA &H7F40,&H4CF3,&H74F0,&H4DF3,&H7F80,&H4EF3,&H78F0,&H60F4
310 DATA &H60F5,&H3F87,&H38F6,&HA41,&H54,&H241,&H940,0
320 DATA ,(BC),DE,(DE),HL,(HL),AF,AF'
330 DATA A,SP,(SP),NZ,Z,NC,C,
```

リスト1 PC-ASM プログラム

```
340 DATA ,,,,I,D,IR,DR
350 DATA NEG,RETN,IM   0,"LD   I,A",RETI,IM   1,"LD   A,I",IM   2,RRD,RLD
360 UA="  1  3  9  B  C1 C5 ":UB="  0  C9 D9 F3 FB 76 ":UC="  CB DD ED FD "
370 UD="2  8  A  10 12 18 1A 20 22 28 2A 30 32 38 3A "
380 UD=UD+"C3 CD D3 DB E3 E9 EB F9 "
390 UE="  44 45 46 47 4D 56 57 5E 67 6F ":Z=58368!:L=0:LL=0
400 DEFFNX(S)=VAL("&h"+S)-INT(VAL("&h"+S)/65536!)*65536!
410 DEFFNI(X)=X-INT(X/256)*256
420 DEFFNO(S)=MID$("0",LEN(S))+S:DEFFNP(S)=MID$("000",LEN(S))+S
430 DIM T(5,7),Y(15),Y2(23),TT(15),TL(30),XL(30)
440 FORI=0TO5:FORJ=0TO7:READT(I,J):NEXT:NEXT
1000 '*MAIN
1010 PRINT:PRINT "READY":INPUT S:IF S="e" THEN END
1030 ON INSTR("pad",S) GOSUB 3420,4000,1050
1040 GOTO 1010
1050 RESTORE260:CA=0
1060 FORI=0TO15:READ Y(I):NEXT:FORI=0TO7:READ Y1(I):NEXT
1070 FORI=0TO23:READ Y2(I):NEXT:FORI=0TO15:READ TT(I):NEXT
1080 FORI=0TO7:READ TU(I):NEXT:FORI=1TO10:READ SU(I):NEXT
1100 INPUT"START,END";TA,TB:S=TA:GOSUB4050:W=S:GOSUB6000:IF ER=1 THEN RETURN ELS
E XS=XN:S=TB:GOSUB4050:W=S:GOSUB6000:IF ER=1 THEN RETURN ELSE XE=XN:X=XS:PRINT
2000 '*D-AS
2010 XX=X:S="LD":C2=0:TA="":TB="":TT="":GOSUB3150:K=KK:A=KAND7:B=K/8AND7
2020 IF K<&H40 OR K>&H7F THEN 2040 ELSE IF K=&H76 THEN S="HALT":GOTO 3000
2030 TA=T(1,B):TB=T(1,A):GOTO 3000
2040 IF K>127 AND K<192 THEN S=MID$(O(1),1+B*4,4):TA=T(5,B):TB=T(1,A):GOTO 3000
2050 Y=Y(8*(K¥128)+A):IF Y=0 THEN 2100 ELSE IF Y=8 THEN 2080
2060 GOSUB 3050:GOSUB 3120:IF C<8 THEN S=MID$(O(3),1+C*4,4) ELSE S=MID$(O(C-10),
1+B*4,4)
2070 GOTO 3000
2080 M=K AND &HCF:Y=Y1(INSTR(UA,HEX$(M)+" ")/3):IF Y=0 THEN 2100
2090 GOSUB 3050:B=(K/16)AND7:GOSUB 3120:S=MID$(O(4),1+C*4,4):GOTO 3000
2100 TK=HEX$(K)+" ":I=INSTR(UC,TK)/3:IF I<>0 THEN 3160
2110 I=INSTR(UB,TK)/3:IF I<>0 THEN S=MID$(O(5),I*4-3,4):GOTO 3000
2120 I=INSTR(UD,TK)/3:Y=Y2(I):GOSUB 3050:S=MID$(O(6),1+C*4,4):TT(15)=TN:TA=TT(CA
):TB=TT(CB)
3000 '*PRINT
3010 IF TA<>"" AND TB<>"" THEN TA=TA+","
3020 XL=XX:TN="":GOSUB3380
3030 PRINT FNP(HEX$(XX));" ";TT;TAB(14);TN;TAB(21);S;TAB(26);TA;TB
3040 IF XX<XE THEN 2000 ELSE RETURN
3050 C2=Y/&H1000:CA=(Y/256)AND15:CB=(Y/16)AND15:C=YAND15
3060 IF C2=0 THEN RETURN ELSE IF C2=1 THEN TA="HL":RETURN
3070 GOSUB 3150:K1=KK:IF C2>5 THEN GOSUB3150:XL=K1+KK*256:GOTO3090
3080 IF C2>3 THEN XL=(K1+128AND255)+XX-126 ELSE TN=FNO(HEX$(K1))+"H":GOTO 3100
3090 TN=HEX$(XL)+"H":IF XL<>0 THEN GOSUB3380
3100 IF C2MOD2=1 THEN TN="("+TN+")"
3110 RETURN
3120 IF CA=0 THEN 3130 ELSE IF CA=15 THEN TA=TN ELSE TA=T(CA,B)
3130 IF CB=0 THEN 3140 ELSE IF CB=15 THEN TB=TN ELSE TB=T(CB,B)
3140 RETURN
3150 KK=PEEK(X):X=X+1:TT=TT+FNO(HEX$(KK)):RETURN
3160 GOSUB3150:K=KK:B=K/8AND7:A=KAND7:B2=B/2:B3=K/64
3170 ON I GOTO 3180,3290,3200,3290
3180 IF B3<>0 THEN S=T(0,B3):TA=HEX$(B):TB=T(1,A):GOTO 3000
3190 IF B=6 THEN 3290 ELSE S=MID$(O(8),1+B*4,4):TB=T(1,A):GOTO 3000
3200 ON B3+1 GOTO 3290,3210,3280,3290
3210 IF A>3 THEN 3270 ELSE IF K=&H63 OR K=&H6B OR K=&H70 OR K=&H71 THEN 3290
3220 IF A=0 THEN S="IN":TA=T(1,B):TB="(C)":GOTO 3000
3230 IF A=1 THEN S="OUT":TA="(C)":TB=T(1,B):GOTO 3000
3240 IF A=3 THEN 3260 ELSE IF BMOD2=0 THEN S="SBC" ELSE S="ADC"
3250 TA="HL":TB=T(4,B2):GOTO 3000
3260 C2=7:GOSUB3070:TA=TN:TB=T(4,B2):IF BMOD2=0 THEN 3000 ELSE TA=TB:TB=TN:GOTO3
000
3270 I=INSTR(UE,HEX$(K)+" ")/3:IF I=0 THEN 3290 ELSE S=SU(I):GOTO 3000
3280 IF B<4 OR A>3 THEN3290 ELSE S=T(0,A+4)+TU(B):GOTO 3000
3290 S="???":TA="??":TB="??":GOTO 3000
3300 '*LBL STORE
3310 IF L+LL>30 THEN PRINT "LBL-MAX":RETURN ELSE I=0:IF L=0 THEN 3340
3320 FOR I=0TOL-1:IF TL(I)=S THEN 3360 ELSE NEXT
```

リスト1　PC-ASM プログラム

```
3330 FOR I=0TOL-1:IF XL(I)<X THEN NEXT
3340 J=L:TL(J)=S:XL(J)=X:L=L+1
3350 IF J>I THEN SWAP TL(J),TL(J-1):SWAP XL(J),XL(J-1):J=J-1:GOTO 3350 ELSE RETU
RN
3360 PRINT "LBL-DBL":RETURN
3380 '*LBL SRC
3390 IF L=0 THEN RETURN ELSE IF XL>XL(L/2) THEN D=L/2+1:G=L-1 ELSE D=0:G=L/2
3395 IF D>G THEN RETURN
3400 FOR I=DTOG:IF XL=XL(I) THEN TN=TL(I):RETURN ELSE NEXT
3410 RETURN
3420 '*LBL PRI
3430 PRINT"LABEL=";L:FOR I=0TOL-1:PRINT TL(I);": ";HEX$(XL(I));"H",:NEXT:PRINT:R
ETURN
4000 '** AS **
4010 FORI=1TO10:W(I)="":NEXT:PRINT "*";FNP(HEX$(Z));" ";FNO(HEX$(PEEK(Z)));" ";:
CA=0
4020 C=1:LINEINPUT T:J=LEN(T):IF T="" THEN Z=Z+1:GOTO 4010
4030 FOR I=1TOJ:W=MID$(T,I,1):IF W=" " OR W="," THEN C=C+1 ELSE IF W="'" THEN 40
70 ELSE GOSUB4060:W(C)=W(C)+W
4040 IF C>10 THEN 5040 ELSE NEXT:GOTO 4070
4050 FOR I=1TOLEN(S):W=MID$(S,I,1):GOSUB4060:MID$(S,I,1)=W:NEXT:RETURN
4060 M=ASC(W):IF M<97 OR M>123 THEN RETURN ELSE W=CHR$(M-32):RETURN
4070 X=Z:W="    ":MID$(W,1)=W(2):ER=0:G=0:D=0:J=0:A=1:B=3:D=0
4080 I=INSTR("ORG EQU END DS  DB  DC  ",W):IF I=0 THEN 4210 ELSE W=W(3)
4090 ON I/4+1 GOTO 4100,4110,4130,4120,4180,4200
4100 GOSUB6000:Z=XN:X=Z:GOTO 5000
4110 GOSUB6000:X=XN:GOTO 5000
4120 GOSUB6000:Z=Z+XN:GOTO 5000
4130 IF LL=0 THEN RETURN ELSE W=TL(31-LL):X=XL(31-LL):GOSUB6060:IF ER=1 THEN PRI
NT"UNDF ";W,HEX$(X):GOTO 4170
4150 K=PEEK(X):Z=X+1:IF KMOD8=0 THEN D=FNI(XN-Z-1) ELSE D=D1:GOSUB5050:D=D2
4160 GOSUB5050
4170 LL=LL-1:GOTO 4130
4180 FOR A=3TOC:W=W(A):GOSUB6000:IF ER=1 THEN5040 ELSE D=D1:GOSUB5050:IF D2<>0 T
HEN D=D2:GOSUB5050
4190 NEXT:GOTO 5030
4200 A=LEN(W(1))+6:FOR I=A TO LEN(T)-1:D=ASC(MID$(T,I)):GOSUB5050:NEXT:GOTO 5030

4210 G=1:IF W(3)>"" OR W(4)>"" THEN 4240
4220 I=INSTR(O(2),W):IF I>0 THEN I=I/4:D=7+8*I:GOTO 5000
4230 I=INSTR(O(5),W):IF I=0 THEN4550 ELSE I=I/4:D=FNX(MID$(UB,I*3+3)):GOTO 5000
4240 IF W<>"LD  " THEN 4310 ELSE GOSUB4570:IF N=8 THEN 4280
4250 D=N*8+6:B=4:GOSUB4570:IF N<8 THEN D=58+D+N:GOTO 5000
4260 G=2:W=W(4)
4270 GOSUB6000:IF ER=0 THEN 5000 ELSE IF D<>62 THEN 5040
4280 W=W(3)+W(4):G=1:I=INSTR(O(7),W):IF I<>0 THEN I=I/5:ER=0:D=2+8*I:IF I<4 THEN
 5000 ELSE D=249:GOTO 5000
4290 CA=1:G=3:W=LEFT$(W(3),1)+LEFT$(W(4),1):I=INSTR("(H(A(",W):IF I>0 THEN D=26+
8*I:B=4-IMOD2:W=MID$(W(B),2,LEN(W(B))-2):GOTO 4270
4300 A=4:GOSUB4570:IF N>3 THEN 5040 ELSE D=1+16*N:W=W(4):GOTO 4270
4310 M=INSTR(O(1),W):IF M>0 THEN M=M/4 ELSE M=INSTR(O(8),W):IF M=0 THEN 4380 ELS
E M=M/4:G=2:D=203:GOTO 4360
4320 IF W(4)>""THEN SWAP W(3),W(4)
4330 IF W(4)<>T(5,M) THEN IF M>0 OR W(4)<>"HL" THEN 5040 ELSE K=9:GOTO 4520
4340 GOSUB4570:IF N<8 THEN D=128+8*M+N:GOTO 5000
4350 W=W(3):GOSUB6000:D=198+8*M:G=2:GOTO 5000
4360 IF M<8 THEN D1=8*M ELSE D1=64*(M-7)+8*VAL(W(3)):W(3)=W(4)
4370 GOSUB4570:IF N=8 THEN 5040 ELSE D1=D1+N:GOTO 5000
4380 W=W(2):CA=1:IF W="RET" THEN A=2 ELSE IF W="RST" THEN A=3 ELSE 4400
4390 GOSUB4570:IF N=8 THEN5040 ELSE D=178+A*7+8*N:GOTO 5000
4400 IF W="JP" THEN K=194:D=195 ELSE IF W="CALL" THEN K=196:D=205 ELSE 4450
4410 W=W(3):IF W(4)>"" THEN 4440
4420 IF K=194 AND W="(HL)" THEN D=233:GOTO 5000
4430 GOSUB6000:G=3:GOTO 5000
4440 A=2:GOSUB4570:IF N=8 THEN 5040 ELSE W=W(4):D=K+N*8:GOSUB6000:G=3:GOTO 5000
4450 IF W="INC" THEN K=3 ELSE IF W="DEC" THEN K=11 ELSE 4470
4460 GOSUB4570:IF N=8 THEN 4520 ELSE D=4+K/8+8*N:GOTO 5000
4470 IF W="DJNZ" THEN D=16:GOTO 4500
4480 IF W<>"JR" THEN 4510 ELSE IF W(4)="" THEN D=24:GOTO 4500
4490 A=2:GOSUB4570:IF N>3 THEN 5040 ELSE D=32+8*N:B=B+1
```

リスト1 PC-ASM プログラム

```
4500 W=W(B):GOSUB6000:XN=XN-Z-2:IF XN>-129 AND XN<128 THEN D1=FNI(XN):D2=0:G=2:G
OTO 5000:ELSE PRINT "ER-OVR":GOTO 4010
4510 J=4:IF W="POP" THEN K=129 ELSE IF W="PUSH" THEN K=133 ELSE 4530
4520 A=4:GOSUB4570:IF N=8 THEN5040 ELSE D=K+N*16:GOTO 5000
4530 IF W="EX" THEN W=W(3)+W(4):I=INSTR("DEHL(SP)HLAFAF'",W):IF I=0 THEN 5040 EL
SE I=I/5:D=VAL(MID$("2352278",I*3+1,3)):GOTO 5000
4540 IF W="OUT" THEN D=211:SWAP W(3),W(4):GOTO 4560
4550 IF W="IN" THEN D=219 ELSE PRINT "ER-INS":GOTO 4010
4560 IF W(3)<>"A" THEN 5040 ELSE W=MID$(W(4),2,LEN(W(4))-2):CA=0:GOSUB6000:G=2:G
OTO 5000
4570 FOR N=JTO7:IF T(A,N)=W(B) THEN RETURN ELSE NEXT:RETURN
5000 '*DATA STORE
5010 IF ER=1 THEN 5040 ELSE IF G=0 THEN 5030 ELSE IF G=2 AND D2<>0 THEN 5040 ELS
E GOSUB5050
5020 IF G>1 THEN D=D1:GOSUB5050:IF G>2 THEN D=D2:GOSUB5050
5030 S=W(1):IF S>""THEN IF S>="A" AND S<"[" AND LEN(S)<7 THEN GOSUB3310 ELSE PRI
NT "ER-LBL"
5035 GOTO 4010
5040 PRINT "ER-OP":GOTO 4010
5050 POKE Z,D:Z=Z+1:RETURN
6000 '*CHR-NUM
6010 IF W="" THEN 6100 ELSE T=LEFT$(W,1):IF T<"0" THEN 6100 ELSE IF T>"9" THEN 6
040
6020 IF RIGHT$(W,1)<>"H" THEN XN=VAL(W):GOTO6090 ELSE IF ASC(W)=48 THEN W=MID$(W
,2)
6030 IF LEN(W)>5 THEN 6100 ELSE XN=FNX(W):GOTO 6090
6040 '*LBL-NUM
6050 IF T<"A" OR T>="[" THEN 6100 ELSE I=0
6060 I=0
6070 IF L>I THEN IF TL(I)=W THEN XN=XL(I):GOTO 6090 ELSE I=I+1:GOTO 6070
6080 IF CA=0 OR LEN(W)>6 THEN 6100 ELSE IF L+LL>30 THEN PRINT "LBL-MAX":GOTO 610
0 ELSE TL(30-LL)=W:XL(30-LL)=Z:LL=LL+1:XN=Z
6090 IF XN<65536! THEN D1=FNI(XN):D2=XN/256:ER=0:RETURN
6100 ER=1:RETURN
```

●テキストがリアルタイムでエディト，アセンブルできる！

2パス・アセンブラ

PC-8001版

佐々木 哲哉

PC-8001用の2パス・アセンブラを作ったので発表します．

このアセンブラは『システム・プログラム・ライブラリ』に発表したMZ用のアセンブラにアセンブル時のメモリ・アドレス指定，画面およびプリンタのリスティング，エディタのサーチ・コマンドなどを付加したものです．

また，このアセンブラの最大の特長はすべてをメモリ上で行なうので，機械語の開発がインタープリタのように行なえることです．これは，煩わしいカセット・テープへのやり取りをなくし，メモリ上にアセンブラ，エディタ，ソース，オブジェクトを同居させて，『プログラム→アセンブル→ＲＵＮ』という型式を取っているからです．

これで，『機械語を作る』という感覚でなく，『インタープリタを使う』という感覚でプログラムが作れます．

ですから，いまから機械語の勉強をして機械語のプログラムをしようとしているのであれば，このアセンブラはきっと機械語勉強の役に立つことでしょう．また，機械語を一応知っていて，ハンド・アセンブルをしているのであれば，このアセンブラのインタープリタ的感覚と便利さに驚かれることでしょう．

アセンブラの仕様

システム・プログラム ライブラリで発表したMZ用アセンブラとほとんど同じですが，マクロ命令やコマンドが少々異なります．ニモニックはインテル型式です．インテル型式に慣れていない人のために，ザイログ-インテル・ニモニック対照表を表1と表2に示しておきますので見てください．

1 ニモニック

❶8080の命令は表2を参照してください．

❷Z80の命令は次のようになっています．

●ザイログ型式のニモニックと同じ命令は次のとおりです．

EX, JR, DJNZ, EXX, LDI, LDIR, LDD, LDDR, CPIR, CPD, CPDR, NEG, BIT, RES, SET.

●ザイログ型式のニモニックと異なる命令は表3を参照してください．

❸擬似・マクロ命令は表4を参照してください．

表1-a ザイログ・ニモニック—マシン語対応表

下位＼上位	0	1	2	3	4	5	6	7
0	NOP	DJNZ e	JR NZ, e	JR NC, e	LD B, B	LD D, B	LD H, B	LD (HL), B
1	LD BC, nn	LD DE, nn	LD HL, nn	LD SP, nn	LD B, C	LD D, C	LD H, C	LD (HL), C
2	LD (BC), A	LD (DE),A	LD (nn), HL	LD (nn), A	LD B, D	LD D, D	LD H, D	LD (HL), D
3	INC BC	INC DE	INC HL	INC SP	LD B, E	LD D, E	LD H, E	LD (HL), E
4	INC B	INC D	INC H	INC (HL)	LD B, H	LD D, H	LD H, H	LD (HL), H
5	DEC B	DEC D	DEC H	DEC (HL)	LD B, L	LD D, L	LD H, L	LD (HL), L
6	LD B, n	LD D, n	LD H, n	LD (HL), n	LD B, (HL)	LD D, (HL)	LD H, (HL)	HALT
7	RLCA	RLA	DAA	SCF	LD B, A	LD D, A	LD H, A	LD (HL), A
8	EX AF, AF'	JR e	JR Z, e	JR C, e	LD C, B	LD E, B	LD L, B	LD A, B
9	ADD HL, BC	ADD HL, DE	ADD HL, HL	ADD HL, SP	LD C, C	LD E, C	LD L, C	LD A, C
A	LD A, (BC)	LD A, (DE)	LD HL, (nn)	LD A, (nn)	LD C, D	LD E, D	LD L, D	LD A, D
B	DEC BC	DEC DE	DEC HL	DEC SP	LD C, E	LD E, E	LD L, E	LD A, E
C	INC C	INC E	INC L	INC A	LD C, H	LD E, H	LD L, H	LD A, H
D	DEC C	DEC E	DEC L	DEC A	LD C, L	LD E, L	LD L, L	LD A, L
E	LD C, n	LD E, n	LD L, n	LD A, n	LD C, (HL)	LD E, (HL)	LD L, (HL)	LD A, (HL)
F	RRCA	RRA	CPL	CCF	LD C, A	LD E, A	LD L, A	LD A, A

表1-b　ザイログ・ニモニック―マシン語対応表

下位＼上位	8	9	A	B	C	D	E	F
0	ADD A, B	SUB B	AND B	OR B	RET NZ	RET NC	RET PO	RET P
1	ADD A, C	SUB C	AND C	OR C	POP BC	POP DE	POP HL	POP AF
2	ADD A, D	SUB D	AND D	OR D	JP NZ,nn	JP NC, nn	JP PO, nn	JP P, nn
3	ADD A, E	SUB E	AND E	OR E	JP nn	OUT(n), A	EX (SP), HL	DI
4	ADD A, H	SUB H	AND H	OR H	CALL NZ, nn	CALL NC, nn	CALL PO,nn	CALL P,nn
5	ADD A, L	SUB L	AND L	OR L	PUSH BC	PUSH DE	PUSH HL	PUSH AF
6	ADD A, (HL)	SUB (HL)	AND (HL)	OR(HL)	ADD A, n	SUB n	AND n	OR n
7	ADD A, A	SUB A	AND A	OR A	RST 0	RST 10H	RST 20H	RST 30H
8	ADC A, B	SBC A, B	XOR B	CP B	RET Z	RET C	RET PE	RET M
9	ADC A, C	SBC A, C	XOR C	CP C	RET	EXX	JP (HL)	LD SP,HL
A	ADC A, D	SBC A, D	XOR D	CP D	JP Z, nn	JP C, nn	JP PE, nn	JP M, nn
B	ADC A, E	SBC A, E	XOR E	CP E	SHIFT & BIT	IN A, (n)	EX DE, HL	EI
C	ADC A, H	SBC A, H	XOR H	CP H	CALL Z, nn	CALL C, nn	CALL PE,nn	CALL M,nn
D	ADC A, L	SBC A, L	XOR L	CP L	CALL nn	IX関係	EXT	IY関係
E	ADC A, (HL)	SBC A(HL)	XOR (HL)	CP (HL)	ADC A, n	SBC A, n	XOR n	CP n
F	ADC A, A	SBC A, A	XOR A	CP A	RST 8	RST 18H	RST 28H	RST 38H

表2-a　本アセンブラのニモニック―マシン語対応表

下位＼上位	0	1	2	3	4	5	6	7
0	NOP	DJNZ e	JRNZ e	JRNC e	MOV B,B	MOV D, B	MOV H, B	MOV M, B
1	LXI B␣nn	LXI D␣nn	LXI H␣nn	LXI SP␣nn	MOV B, C	MOV D, C	MOV H, C	MOV M, C
2	STAX B	STAX D	SHLD nn	STA nn	MOV B, D	MOV D, D	MOV H, D	MOV M, D
3	INX B	INX D	INX H	INX SP	MOV B, E	MOV D, E	MOV H, E	MOV M, E
4	INR B	INR D	INR H	INR M	MOV B, H	MOV D, H	MOV H, H	MOV M, H
5	DCR B	DCR D	DCR H	DCR M	MOV B, L	MOV D, L	MOV H, L	MOV M, L
6	MVI B␣n	MVI D␣n	MVI H␣n	MVI M␣n	MOV B, M	MOV D, M	MOV H, M	HLT
7	RLC	RAL	DAA	STC	MOV B, A	MOV D, A	MOV H, A	MOV M, A
8	EX	JR e	JRZ e	JRC e	MOV C, B	MOV E, B	MOV L, B	MOV A, B
9	DAD B	DAD D	DAD H	DAD SP	MOV C, C	MOV E, C	MOV L, C	MOV A, C
A	LDAX B	LDAX D	LHLD nn	LDA nn	MOV C, D	MOV E, D	MOV L, D	MOV A, D
B	DCX B	DCX D	DCX H	DCX SP	MOV C, E	MOV E, E	MOV L, E	MOV A, E
C	INR C	INR E	INR L	INR A	MOV C, H	MOV E, H	MOV L, H	MOV A, H
D	DCR C	DCR E	DCR L	DCR A	MOV C, L	MOV E, L	MOV L, L	MOV A, L
E	MVI C␣n	MVI E␣n	MVI L␣n	MVI A␣n	MOV C, M	MOV E, M	MOV L, M	MOV A, M
F	RRC	RAR	CMA	CMC	MOV C, A	MOV E, A	MOV L, A	MOV A, A

表2-b　本アセンブラのニモニック―マシン語対応表

下位＼上位	8	9	A	B	C	D	E	F
0	ADD B	SUB B	ANA B	ORA B	RNZ	RNC	RPO	RP
1	ADD C	SUB C	ANA C	ORA C	POP B	POP D	POP H	POP PSW
2	ADD D	SUB D	ANA D	ORA D	JNZ nn	JNC nn	JPO nn	JP nn
3	ADD E	SUB E	ANA E	ORA E	JMP nn	OUT n	XTHL	DI
4	ADD H	SUB H	ANA H	ORA H	CNZ nn	CNC nn	CPO nn	CP nn
5	ADD L	SUB L	ANA L	ORA L	PUSH B	PUSH D	PUSH H	PUSH PSW
6	ADD M	SUB M	ANA M	ORA M	ADI n	SUI n	ANI n	ORI n
7	ADD A	SUB A	ANA A	ORA A	RST 0	RST 2	RST 4	RST 6
8	ADC B	SBB B	XRA B	CMP B	RZ	RC	RPE	RM
9	ADC C	SBB C	XRA C	CMP C	RET	EXX	PCHL	SPHL
A	ADC D	SBB D	XRA D	CMP D	JZ nn	JC nn	JPE nn	JM nn
B	ADC E	SBB E	XRA E	CMP E	SHIFT & BIT	IN n	XCHG	EI
C	ADC H	SBB H	XRA H	CMP H	CZ nn	CC nn	CPE nn	CM nn
D	ADC L	SBB L	XRA L	CMP L	CALL	IX	EXT	IY
E	ADC M	SBB M	XRA M	CMP M	ACI n	SBI n	XRI n	CPI n
F	ADC A	SBB A	XRA A	CMP A	RST 1	RST 3	RST 5	RST 7

以上が本アセンブラのアセンブルできる命令です．これだけでは，多少説明不足なので，わかりにくい命令について１つずつ説明します．

1)ＯＦＳＥＴ命令
この命令はアセンブラが命令を解読して，実際にメモリに落とすときのアドレスを指定する命令です．たとえば，ＯＦＳＥＴ　＄Ｄ000とすれば＄Ｄ000からアセンブルした結果を順次ストアしていきます．だから，必ずソースの先頭にこの命令を入れてください．

2)ＯＲＧ命令
アセンブル開始アドレスを指定する命令です．この命令のオペランドにはメモリのないアドレスも指定できるので，ＲＯＭの開発もできます．

3)＝命令
ラベルの値を定義する命令です．ラベルでラベルを定義する場合は，すでに定義されていなければなりません．

4)！命令
この命令は実に便利な命令で，16進*n*桁をストアしてくれます．たとえば，！010000とすればＬＸＩ␣＄0000と同じで，ソースを２バイト分だけ短かくできます．また，省メモリの他にデータをずらずらと書き並べることもできます．なお，このアセンブラでアセンブルできない命令はこの命令を使って書くことができるので有効に使ってください．

5)その他の命令
その他の命令は各自アセンブルしてアセンブル・リストを見て，どのようにアセンブルされているか確かめてください．

2 ラベル

❶行の先頭にあること．
❷『＄』，『：』，『０』～『９』で始まらないこと．
❸スペース，『＋』，『－』を含まないこと．
❹文字数に制限はないが７文字以上であるとアセンブル・リストが見づらくなる．
❺ラベルの後に『：』は付けなくて良い．

3 オペランド

1)ニモニックとオペランド間の区切りはスペースであること．

2)ＭＯＶ，ＢＩＴ，ＲＥＳ，ＳＥＴ命令のオペランド間の区切りのみコンマで，他はスペースであること．

3)スペースはいくら入れても良い．

4)２バイト命令のオペランドにダブル・クォテーションでスペースは定義できない．また，ダブル・クォテーションを定義するときは次のように２つ並べれば良い（ＭＶＩ␣Ａ␣" "）．
5)ラベルの最後に10進加減算が可能．
●２バイト命令のオペランド

表３　ザイログ型式のニモニックと異なる命令

ザイログ	本アセンブラ
JR␣NZ	ＪＲＮＺ相対ジャンプ命令はインテルのＪＺ，ＪＮＺと同じようにスペースを入れない．
IX　IY	ＩＸはＤＤをストア ＩＹはＦＤをストア．
LD (nn), rr'	ＳＸＤ␣ｒ␣nn 例）ＳＸＤ␣Ｄ␣＄Ｄ０００
LD rr',(nn) SBC HL, rr' ADC HL, rr' CPI	ＬＸＤ␣ｒ␣nn 例）ＬＸＤ␣Ｂ␣＄Ｄ０００ ＳＢＣ␣ｒ　例）ＳＢＣ␣Ｂ ＡＤＸ␣ｒ　例）ＡＤＸ␣Ｂ ＣＰＩＮ

表４　擬似・マクロ命令

！ ＝ ； ＃ ＯＲＧ ＭＳＧ"文字列" ＤＷ␣ｎ ｎ	16進*n*バイトをストア　例）！ＣＤ1800 ラベルを定義する．例）ＬＰ␣＝␣＄12 コメント　例）；コメントダ！ 10進１バイトをストア　例）＃128 アセンブル開始アドレスを指定する． 例）ＯＲＧ␣＄0000 クォーテーション内の文字列をＡＳＣＩＩコードでストアする． 例）ＭＳＧ␣"ＡＳＳＥＭＢＬＥＲ" オペランドをＬ，Ｈの順でストアする． 例）ＤＷ␣＄1234
ＭＶＸ␣ｒ␣ｒ	レジスタ・ペアのＭＯＶ命令 例）ＭＶＸ␣Ｈ␣Ｄ
ＳＬＸ␣ｒ ＳＲＸ␣ｒ	レジスタ・ペアの左方向シフト [c]←[15　0]←0 例）ＳＬＸ　Ｂ レジスタ・ペアの右方向シフト 0→[15　0]→[c]　例）ＳＲＸ　Ｂ
ＳＴＸ ＬＤＸ	（ＤＥ）←ＨＬ ＨＬ←（ＤＥ）
ＰＲＩＮＴ "文字列" ＧＥＴＫＹ ＩＮＫＥＹ ＧＥＴＬＮ ＣＲ ＬＦ	クォーテーション内をプリントする． 例）ＰＲＩＮＴ␣"プリント" １文字入力 リアルタイム１文字入力 １行入力 キャリッジ・リターン ライン・フィード
ＬＡＡ　＊ ＬＡＨ　＊	Ａレジスタに（ＤＥ）からの16進ASCIIコードを２進にして入れる．エラー➡CY=1 HLレジスタに（DE）からの16進ASCIIコードを2進にして入れる．エラー➡CY=1
ＰＲＡ　＊ ＰＲＨ　＊	Ａレジスタをプリント ＨＬレジスタをプリント
ＴＳＴ␣ｒ ＴＳＸ␣ｒ ＣＬＲ␣ｒ ＣＬＸ␣ｒ	レジスタが０かテストする． レジスタ・ペアが０かテストする． レジスタをクリアする． レジスタ・ペアをクリアする．
ＯＦＳＥＴ␣nn	コンピュータ内のメモリに落とす実際のアドレス指定．

注意）＊の付いている命令はこのアセンブラがないと動作しません．

ダブル・クォーテーション	"	ＡＳＣＩＩ
ドル・マーク	＄	16進
数字	０～９	10進
＃ラベル		ラベル上位
ラベル		ラベル下位

●３，４バイト命令のオペランド

数字	0～9	10進
ドル・マーク	$	16進
ラベル		ラベル

4 エラー

1)MNEM ERROR
ニモニックのつづり間違い，またはニモニックの先頭にスペースがないなど．

2)OP ERROR
オペランドがその命令のニモニックに合わない．オペランドの書き忘れなど．

3)UNDEF LABEL ERROR
ラベルの書き忘れ，つづり間違いなど．

4)LABEL NAME ERROR
ラベルの先頭に『0』～『9』があった．『＋』，『－』が含まれていたなど．

5)JMP ERROR
相対ジャンプの範囲を越えている．

6)LABEL TABLE FULL
ラベル・テーブルがメモリのない範囲にきた．

7)＊MEMORY FULL
エディタのエラーで，テキスト・エリアがFULLになったこと．インサート・モードで，このエラーが出たら，STOP キーを押すこと．

8)Tape read error
これもエディタ（ROM）のエラーです．これはテープの読み間違い，STOP キーが押されたなどです．このエラーが出たらN-BASICに戻っていますから，**mon** CR でホット・スタートにジャンプさせてください．

上記1)～5)のエラーは文番号とその行を表示して止まります．エディタにしてその行を書き直してください（LPはエラーの行にある）．

5 アセンブラ・コマンド

モニタはN-BASICのROMをコールしています．戻るときは CTRL B を入力してください．

テープのロード，セーブ，メモリの読み書きはこれで行なってください．

■各コマンド使用上の注意

1)各コマンドは『＊』の出ているとき入力し，カーソルの上下の移動はしないでください．

2)プリンタ・モードにしてプリンタが接続されてない場合は，STOP キーを入力してください．

3)**O**コマンドはラベルのあるソースをアセンブルした後で使用してください．

4)**M**, **O** コマンドは出力中，キー入力があるとストップします．好きなキーで続きが見れます（**B**コマンドは スペース キーがストップ）．

アセンブラ・コマンド表を**表5**に示しておきます．

表5 アセンブラ・コマンド

コマンド	機能および使用法	コマンドのストップ法
＊A ＊B	アセンブル アセンブル ＆ リスト ＊AまたはB LABEL NAME FROM？ E000	STOP
＊Mm	mからメモリ・ダンプ	X 大文字
＊J m	mへジャンプ	
＊！	モニタをコール	CTRL B
＊E	エディタをコール	＃ R RET
＊O	ラベル・テーブルを出力	X
＊＃	プリンタ・モードの切り換え	
＊P	入力行をプリンタ出力	RET

6 パス

❶PASS0
ラベルをバッファにストア

❷PASS1
ラベルを定義しながらアセンブル

❸PASS2
定義されたラベルでアセンブル．このとき**B**コマンドであればリストを出力し，プリンタ・モードであれば，プリンタにも出力する．

PASS0～2はアセンブラが自動的に切り換えます．

7 その他の規則

❶ラベルの2重定義をしない(エラーがでない)．
❷OFSET命令を必らず先頭に入れること．
❸マルチ・ステートメントが可能．
❹アセンブルするときは，ソース・プログラムとオブジェクト・プログラム，およびラベル・テーブルが重ならないように注意してください．
❺JR命令はラベルで定義してください．
【例】 LOOP DJNZ LOOP
❻ニモニックの先頭はスペースであること．

2 エディタの仕様

MZ用に作られていたものをPC用にしただけで，ほとんど変わっていません．N-BASICでスクリーン・エディットを使える人であれば，だれでもすぐに使いこなせると思います．

1 エディタ・コマンド

表6にエディタのコマンドを示しておきます．

2 エディタ使用上の注意

1)タイプ・アウト中にキー入力があると，次にキー入力があるまで停止します．このキー入力のとき，X を入力すればコマンド待ちになります．他のキーはタイプを続けます．

2)エディットはBASIC同様にカーソルを使って書き換え，RET を入力すればエディットできます．

3) インサート・モードではカーソル・エディットできません．このモードを抜けるには，RETキーのみ入力かSTOPキーを入力してください．
4) プログラムをテープから結合する場合は，まず，・コマンドと，コマンドでテキストのアドレスを調べておき，Xコマンドで・のアドレスにテキスト・ポインタを置き，そこでロードします．
ロードが終わってからXコマンドでテキスト・ポインタを，のアドレスへ戻します．この操作によってRAM上のプログラムの後に，テープのプログラムを結合することができます．
5) Xコマンドでテキスト・ポインタを変更して，新しくプログラムを入力するときは&コマンドでクリアしておいてください．
6) ロード中はSTOPキーを押さないでください．
7) Jコマンドでサブルーチンにジャンプするとアセンブラに戻ります．
8) プリンタの接続されてない場合，プリンタ・モードをセットしないでください．

3 プログラムの入力

このアセンブラは32Kシステム用にアセンブルしているので，16Kの人はメモリを買うか，カセット・サービスをしてもらうと良いと思います．

32Kシステムの人はダンプ・リストから入力してください．

5Kバイト弱のプログラムなので，入力は慎重に行なってください．

4 アセンブラの使用法

アセンブラの使い方を述べる前に1つ重要なことを書いておきます．それはOFSET命令とORG命令の関係です．これは不連続アドレスへのアセンブルのときに，とても重要なことなので頭に入れておいてください．

説明の都合上，例を出して説明します．たとえば，$E000～と$E100～にプログラムを作成したいとします．このとき，PCにはメモリがあるアドレスなので次のようになります．

OFSET　$E000　ORG　$E000
❶ {～命令（アセンブル記述）
OFSET　$E100　ORG　$E100
❷ {～命令（アセンブル記述）

ここで下の方のOFSET命令を取ったらどのようになるかと言いますと，❶のアセンブル終了アドレスに続けて$E100からのプログラムをアセンブルすることになります．だからそのまま走らせることはできません．

これを走らせたいなら❷のプログラムを$E100にブロック転送しなければなりません．逆に今度はORG命令を取ったら，$E100にプログラムがアセンブルされますが，このアドレスで直接走らせられません．走らせるには，リロケートしなければならなくなります．

表6　テキスト・エディタのコマンド表

コマンド	機　　能	LP移動の有無
#T	LPからタイプ	無
#TA	全行タイプ	無
#Tn	n行からタイプ	有
#P	LPをエンドに移動	有
#Pn	LPをn行へ移動	有
#PB	LPを1行目へ移動	有
#D	LPの行を1行削除	無
#Dn	n行目を削除	有
#DPn	LPよりn行削除	無
#I	LPからインサートする．	有
#Wf	セーブ（テキストのみ）	無
#L	ロード（，のアドレスから）	無
#Xm	テキスト・ポインタをmにする．	有
#，	テキストの先頭アドレス表示(テキスト・ポインタ)	無
#・	テキストのエンド・アドレス表示	無
#／	LPの文番号表示	無
#Jm	mへマシン語ジャンプする．	無
#&	ニュー〔NEW〕テキスト・ポインタのアドレスに00をストア	有
#E	エディット・モードにする．	有
#R	テキスト・エディタからリターンする．	無
#Ss	sと言う文字列をサーチし，あればLPをその行へセットし，なければ最終行へセットする．	有
##	プリンタのモード切り換え	無

n＝10進数　m＝16進4桁　f＝ファイル・ネーム（2文字）
s＝文字列　LP＝ライン・ポインタ

注意：テキスト・ポインタと言うのはテキストがどこから始まっているかを示しています．このアドレスを変更すれば，メモリ上に複数個のテキストを置くことができます．そのコマンドがXコマンドです（アセンブラは，テキスト・ポインタのあるプログラムをアセンブルします）．

実行例　1

```
** PC-8001 ASSEMBLER VER 5.0 **
*
```

実行例　2

```
*E
#&
#I
 OFSET $C000 ORG $C000
 MVI A 30
LOOP PUSH PSW
 MVI A "#" CALL $18
 MVI A 7   CALL $18
 POP PSW DCR A JRNZ LOOP
 JMP $8103

#
```

以上のように，OFSET命令とORG命令は密接な関係ですから注意してください．

では，アセンブラの使用法を述べて行きたいと思います．プログラムは入力されていますか？　入力されていれば一応モニタでセーブしてください．それではモニタから＊G 8100 RETと入力してください．実行例1のように表示されコマンド待ちになっています．さあ，アセンブラでのプログラム開始です．

まずはエディタをコールします．E RETと入力してください(小文字も可)．コールド・スタート直後は&コマンドは入力の必要はないのですが，& RETと入力しテキスト

実行例 3

```
#TA
   1  OFSET $C000 ORG $C000
   2  MVI A 30
   3 LOOP PUSH PSW
   4  MVI A "#" CALL $18
   5  MVI A 7  CALL $18
   6  POP PSW DCR A JRNZ LOOP
   7  JMP $8103
#
```

実行例 4

```
#R
*A
LABEL NAME FROM ?A000
PASS 0
PASS 1

PASS 2

*
```

実行例 5

```
*B
LABEL NAME FROM ?AC00
PASS 0
PASS 1

PASS 2
C000                  OFSET $C000
C000                  ORG   $C000
C000 3E1E             MVI   A 30
C002 F5        LOOP   PUSH  PSW
C003 3E23             MVI   A "#"
C005 CD1800           CALL  $18
C008 3E07             MVI   A 7
C00A CD1800           CALL  $18
C00D F1               POP   PSW
C00E 3D               DCR   A
C00F 20F1             JRNZ  LOOP
C011 C30381           JMP   $8103

*
```

実行例 6

```
*MC000

C000 3E 1E F5 3E 23 CD 18 00
C008 3E 07 CD 18 00 F1 3D 20
C010 F1 C3 03 81 00 FF 00 FF
*
```

実行例 7

```
*D
* LABELE TABLE *
LOOP      =$C002
*
```

をクリアします．それから**I** **RET**と入力しインサート・モードにして次々とソース・プログラムを入力していきます．

実行例2を見て入力してください．入力を終わったら，**RET**キーのみ入力するとコマンド待ちになります（入力は小文字でできます．クォーテーションの中はBASICのように必要であれば大文字にします）．

それでは，入力されたものをリストして見ましょう．コマンドは**TA**です．**実行例3**のようにリストされたと思います．ここで誤りがあれば**E** **RET**と入力し，カーソルを使ってＢＡＳＩＣと同様にエディットして**RET**キーを入力してください．下まで**RET**を入力するとコマンド待ちになります．

いよいよアセンブルです．まず**R** **RET**と入力しエディタからアセンブラに制御を移します．**実行例4**のようにコマンドとアドレスを入力すれば，ＰＡＳＳ０～２まで表示してアセンブルを終了します．

ここでもう走らせることができるのですが，一体どのようにアセンブルされているか，**B**コマンドを使ってみることにします．操作は**A**コマンドと同じです．**実行例5**のように表示されます．これでいかにアセンブラが便利なシステム・ソフトであるかわかっていただけると思います．

次にメモリ・ダンプとラベル表示を行ないましょう．**実行例6，7**のように入力してください．メモリ・ダンプは大文字の**X**を入力するとコマンド待ちになります．

一応アセンブルがエラーなく終了したところで，ソース・プログラムをセーブしましょう．まず，**E** **RET**と入力しエディタに制御を移します．そして**W**○○と入力（○はファイル・ネームで，必ず２文字必要）してカセットのＰＬ

写真1　エディタ・モードでテキストを作成

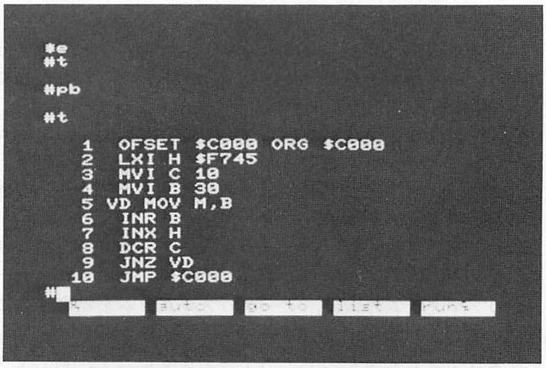


写真2　Bコマンドでアセンブルが完了

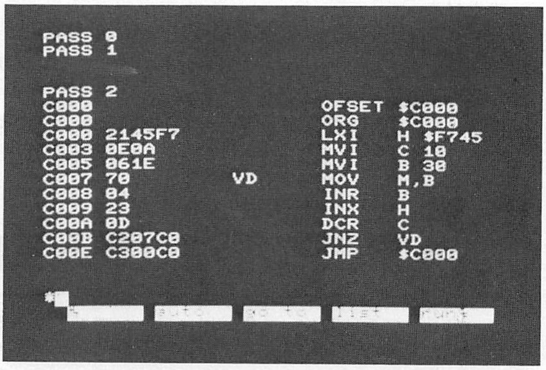
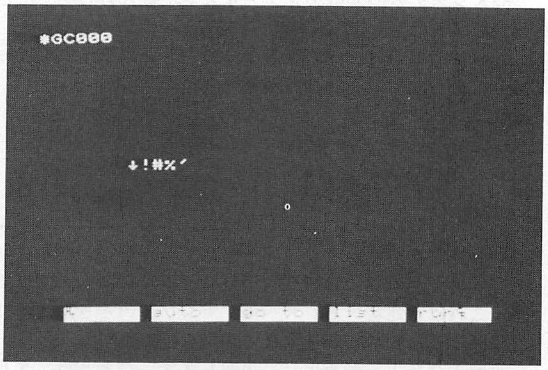


写真3　でき上がったプログラムを実行したところ



AYボタンとRECボタンを押して[RET]キーを入力します．

余談になりますが，ファイル・ネームの6文字は少々中途半端な長さで，6文字ではどうも足りません．MZの16文字でも少々短かいくらいです．どっちにしても短かいのであれば思いきり短かくしようと言うことで2文字にしました．

セーブが終了すると『ピー』と音がしセーブ終了を知らせます．ロードは**実行例9**のようにします．

それでは，こんどはオブジェクトをセーブします．もう一度アセンブラに制御を移して，[!][RET]と入力しROMモニタにします．これはマニュアルに書いてあるモニタですから，そちらの方を良く読んでセーブしてください．**実行例10**に一応示しておきます．

実行例8

```
*E
#WTS

#
```

実行例9

```
#L
LOADING TS
OK
#
```

実行例10

```
#R
*!
*WC000,C014
*
```

5 ついでのお話

私はこのアセンブラを作成するきっかけになったのは，自作マイコン・システムSKY-Z（スカイゼット）のROM開発だったのです．このSKY-ZはCPUにZ80Aを使い，4MHzで動作しています(4.5MHzでも動作した)．CRTディスプレイに秋月のVDGを使った256×192のグラフィックやPSGを使ったシンセサイザなどのハードを生かすソフトは，どうしてもアセンブラに頼らざるを得ません．

しかし，Assembler ver 3.0ではメモリのないアドレスにアセンブルできない，アセンブル・リストがでないなどで，SKY-ZのROMは開発ができませんでした．そこで頭にきて，ver 3.0でアセンブラを記述したわけです．

アセンブル記述のアセンブラだから変更が容易になりました．それでいろいろ付加してMZ用のアセンブラにしました．そこで欲が出て，PC用を作ってみようと思ったのが運の尽きで，PC用のI/Oルーチンやテープのロード・セーブ，プリンタなどを調べてソースの変更を行なわなければならないという悲劇が起こったのでした．

それだけなら良いのですが，できたプログラムをPCに入れる手段がありません．テープのセーブ・ルーチンをMZに入れるしかないと思っていたところへ，I/O'80年10月号に『MZ→PCテープ変換プログラム』が載っていたので，さっそく入力して使ってみました．このプログラムは正に天の助けでした．

運良く基本機能は9月の末日に動作しました．それから，何日もかかってデバッグをしてやっと完成したのでした．まったくPCを持っていない私がなんとMZ上でPCのアセンブラを完成させたのです．PCのユーザーの皆さん，MZのユーザーに負けないように頑張ってください．MZのユーザーの皆さんも負けずに頑張ってください．

SKY-Zは一体どうなったかというと，いま，フルグラフィック上でBASIC SP-5010が動作し，RAMはD-RAM48K，V-RAM 6K，WORK RAM2K，ROM2Kが乗って順調に動いています．

6 後書き

今回のプログラムは今までになく疲れました．何と言っても私がPCを持っていないということが一番疲れた原因だったようです．今度，PC用のソフトを発表するときはPCを買っているかもしれません．

最後におまけとして，『シフト＆ローテート命令早見表』を付けておきますので利用してください．それでは，またの機会に．

参考文献

1)I/O別冊システム・プログラムライブラリ　2)I/O，'80年2，3，10月号　3)ASCII，'80年9月号　4)N-BASIC入門(ASCII出版)

シフト&ローテート命令早見表

ニモニック	動作	レジスタス・タ							
		B	C	D	E	H	L	M	A
RLC	CY 7 0	00	08	10	18	20	28	30	38
RRC	7 0 CY	01	09	11	19	21	29	31	39
RL	CY 7 0	02	0A	12	1A	22	2A	32	3A
RR	7 0 CY	03	0B	13	1B	23	2B	33	3B
SLA	CY 7 0 0	04	0C	14	1C	24	2C	34	3C
SRA	7 0 CY	05	0D	15	1D	25	2D	35	3D
SLI	CY 7 0	06	0E	16	1E	26	2E	36	3E
SRL	7 0	07	0F	17	1F	27	2F	37	3F



金はなく、女のコにはけっして もてず…

PC-8001版2パス・アセンブラ(32K用)

```
8100 C3 AE 8F C3 C3 8F CD 7E 1B 11 96 EC 3E 20 32 43
8110 81 1A FE 22 CC 28 81 B7 28 07 CD 33 81 12 13 18
8120 F0 3E 0D 12 11 96 EC C9 F5 3A 43 81 EE 20 32 43
8130 81 F1 C9 FE 61 D8 FE 7B D0 C5 4F 3A 43 81 47 79
8140 90 C1 C9 00 3E 0D CD A6 40 3E 0A C3 A6 40 3E 20
8150 C3 A6 40 D5 1A FE 0D 28 06 13 CD A6 40 18 F5 D1
8160 C9 3E 07 C3 A6 40 FE 30 D8 FE 3A 38 09 FE 41 D8
8170 FE 47 38 05 37 C9 D6 30 C9 D6 37 C9 1A CD 66 81
8180 D8 C5 07 07 07 07 4F 13 1A CD 66 81 38 02 81 13
8190 C1 C9 D5 CD 7C 81 38 07 67 CD 7C 81 38 01 6F D1
81A0 C9 00 00 0E 00 0C 29 30 FC CD D2 81 E5 21 00 00
81B0 E3 7B 95 5F 7A 9C 57 3F 38 04 EB 19 EB B7 E3 7D
81C0 17 6F 7C 17 67 E3 0D 28 05 CD D1 81 18 E3 E1 EB
81D0 C9 B7 7C 1F 67 7D 1F 6F C9 21 9B EC 06 06 36 20
81E0 2B 10 FB 21 9B EC 36 0D E5 21 0A 00 CD A3 81 7D
81F0 C6 30 E1 2B 77 7B B2 20 EF C9 C5 E5 D5 CD D9 81
8200 CD 44 81 11 97 EC CD 53 81 D1 E1 C1 C9 C5 E5 D5
8210 CD D9 81 EB CD 53 81 D1 E1 C1 C9 C5 D5 44 4D 21
8220 00 00 CB 3A CB 1B 30 01 09 CB 21 CB 10 7A B3 20
8230 F1 D1 C1 C9 21 00 00 1A FE 20 20 03 13 18 F8 FE
8240 30 38 18 FE 3A 30 14 D5 F5 11 0A 00 CD 1B 82 F1
8250 16 00 D6 30 5F 19 D1 13 1A 18 E4 B7 C9 ED 5B 48
8260 FF 1A B7 28 08 FE 0D 28 03 13 18 F5 13 ED 53 48
8270 FF C9 ED 5B 40 FF ED 53 48 FF C9 CD 72 82 CD 50
8280 82 ED 53 40 FF C9 ED 5B 42 FF ED 53 48 FF 44 4D
8290 0B 78 B1 C8 CD 5D 82 0B 1A B7 C8 18 F4 21 FF FF
82A0 22 46 FF 2A 42 FF 22 48 FF 18 03 CD E5 82 2A 48
82B0 FF 7E B7 C8 CD C8 82 CD AC 0F B7 28 F1 CD 75 0F
82C0 CD AC 0F FE 78 C8 18 E6 ED 5B 46 FF 13 ED 53 46
82D0 FF 13 CD FA 81 CD 4E 81 ED 5B 48 FF CD 53 81 CD
82E0 3A 93 C3 5D 82 2A 42 FF 22 48 FF 01 FE FF 03 2A
82F0 48 FF ED 5B 40 FF B7 ED 52 30 05 CD 5D 82 18 EE
8300 ED 43 46 FF C9 11 97 EC CD 34 82 CD 86 82 2A 48
8310 FF 22 40 FF C9 CD 05 83 CD AB 82 C9 2A 42 FF 7E
8320 B7 28 03 23 18 F9 22 44 FF C9 3A 97 EC FE 0D CA
8330 AB 82 FE 41 CA 9D 82 18 DC 13 1A 32 A1 81 13 1A
8340 FE 0D C8 32 A2 81 CD 1C 83 ED 56 42 FF 2A 44 FF
8350 23 B7 ED 52 EB 7B B2 28 38 CD 46 0C 06 0A 3E B9
8360 CD DA 0C 10 FB 7A CD DA 0C 7B CD DA 0C 3A A1 81
8370 CD DA 0C 3A A2 81 CD DA 0C 01 00 D0 0B 78 B1 20
8380 FB 7E CD DA 0C 23 1B 7A B3 20 F6 CD 2E 0C C3 61
8390 81 CD 44 81 11 9A 83 C3 53 81 53 41 56 45 20 45
83A0 52 52 4F 52 0D ED 5B 42 FF 21 FF E9 B7 ED 52 CD
83B0 F3 0B 06 0A CD 88 0C FE B9 20 F7 10 F7 CD 88 0C
83C0 47 CD 88 0C 4F CD 88 0C 32 A1 81 CD 88 0C 32 A2
83D0 81 C5 D5 11 1E 84 CD 53 81 3A A1 81 CD 18 00 3A
83E0 A2 81 CD 18 00 D1 C1 B7 ED 42 EB 38 17 CD 88 0C
83F0 77 5F 23 0B 78 B1 20 F5 CD 31 0C CD 44 81 11 1B
8400 84 C3 53 81 CD 31 0C 11 0E 84 CD 53 81 C9 2A 4D
8410 45 4D 4F 52 59 20 46 55 4C 4C 0D 4F 4B 0D 4C 4F
8420 41 44 49 4E 47 20 0D 13 CD 92 81 D8 22 42 FF CD
8430 1C 83 2A 42 FF 22 40 FF C9 2A 42 FF 36 0D 18 F2
8440 CD E5 82 2A 46 FF 23 23 EB C3 0D 82 2A 42 FF C3
8450 36 91 CD 1C 83 C3 36 91 13 CD 92 81 D8 D1 E9 13
8460 1A FE 42 28 CD C3 05 83 2A 40 FF CD 72 82 CD 50
8470 82 ED 5B 48 FF 1A 77 B7 C8 13 23 18 F8 13 CD 34
8480 82 44 4D 78 B1 C8 0B CD 68 84 18 F7 13 1A FE 50
8490 28 EB FE 0D 28 D2 CD 34 82 CD 86 82 2A 48 FF 22
84A0 40 FF C3 68 84 D5 01 01 00 2A 40 FF 7E B7 28 04
84B0 23 03 18 F8 E5 1A FE 0D 28 04 13 23 18 F7 23 7C
84C0 FE EA 30 20 7E F5 36 0D 7E FE 0D 20 15 F1 77 EB
84D0 E1 ED B8 2A 40 FF D1 1A 77 FE 0D CA 7B 82 23 13
84E0 18 F5 F1 77 11 0E 84 CD 53 81 CD 44 81 E1 D1 C9
84F0 CD 06 81 1A FE 0D C8 CD A5 84 18 F4 2A 40 FF 22
8500 48 FF EB 21 97 EC 42 4B 0A BE 28 12 7E FE 0D 28
8510 0B 1A FE 0D 28 11 21 97 EC 13 18 EA AF C9 7E FE
8520 0D 28 F9 03 23 18 E1 37 C9 13 1A FE 0D C8 CD FC
8530 84 30 08 CD 7B 82 1A B7 C8 18 F3 CD E5 82 CD C8
8540 82 37 C9 CD 06 81 CD 34 82 7C B5 C8 D5 CD 86 82
8550 2A 48 FF 22 4C FF CD 68 84 D1 13 CD A5 84 18 E3
8560 11 C9 85 D5 3E 23 CD A6 40 CD 06 81 1A FE 26 CA
8570 39 84 FE 53 CA 29 85 FE 54 CA 2A 83 FE 4C CA A5
8580 83 FE 57 CA 39 83 FE 2C CA 4C 84 FE 2E CA 52 84
8590 FE 23 CA 27 90 FE 2F CA 40 84 FE 50 CA 5F 84 FE
85A0 4A CA 58 84 FE 58 CA 27 84 FE 44 CA 8C 84 FE 49
85B0 CA F0 84 FE 45 CA 43 85 FE 52 28 0B FE 0D 28 0F
85C0 11 D3 85 CD 53 81 C9 D1 C9 CD 44 81 C3 60 85 D1
85D0 C3 60 85 43 4F 4D 4D 41 4E 44 20 45 52 52 4F 52
85E0 0D 11 E6 85 18 46 4F 50 0D 11 5E 85 18 3E 4D 4E
85F0 45 4D 0D 11 F8 85 18 34 55 4E 44 45 46 20 4C 41
8600 42 45 4C 0D 11 09 86 18 23 4C 41 42 45 4C 20 4E
8610 41 4D 45 0D 11 19 86 18 13 4A 4D 50 0D 11 22 86
8620 18 0A 32 44 45 46 20 4C 41 42 45 4C CD 44 81 CD
8630 53 81 11 44 86 CD 53 81 CD 44 81 CD E5 82 CD C8
8640 82 C3 C3 8F 20 45 52 52 4F 52 0D E5 21 00 00 ED
8650 5B 48 FF 1A FE 30 38 13 FE 3A 30 0F D6 30 29 44
8660 4D 29 29 09 4F 06 00 09 13 18 E8 EB E1 C9 E5 21
8670 00 00 ED 5B 48 FF 13 1A FE 30 38 EF FE 47 30 EB
8680 FE 3A 38 10 FE 41 30 10 18 E1 29 29 29 29 4F 06
8690 00 09 18 E2 D6 30 18 F2 D6 37 18 EE 3A 50 FF B7
86A0 C8 1A FE 23 C2 05 88 13 ED 53 48 FF CD 05 88 5A
86B0 C9 ED 5B 40 FF 1A 13 FE 0D 20 FA ED 53 40 FF C9
86C0 ED 5B 48 FF 1A FE 20 20 03 13 18 F8 ED 53 48 FF
86D0 C9 ED 5B 48 FF 1A FE 20 28 F2 13 18 F6 ED 5B 46
86E0 FF 1A FE 20 28 E6 FE 0D 28 E2 13 18 F4 CD F3 86
86F0 C3 4D 87 CD D1 86 18 C8 0E 00 21 51 91 ED 5B 48
8700 FF 79 FE 68 C8 1A BE 20 08 7E FE 0D C8 23 13 18
8710 F4 7E FE 0D 28 0A 23 7E FE 0D 20 FA 23 0C 18 DD
8720 1A FE 20 C8 18 F6 0E 00 2A 4E FF ED 5B 48 FF 79
8730 FE 08 C8 1A BE 20 08 7E FE 0D C8 23 13 18 F4 7E
8740 FE 0D C8 23 7E FE 0D 20 FA 23 0C 18 DE E5 21 03
8750 93 22 4E FF CD 26 87 E1 C9 E5 21 1D 93 18 F2 E5
8760 21 13 93 18 EC E5 21 26 93 18 E6 79 07 07 07 80
8770 77 23 CD F3 86 1A FE 24 28 11 FE 22 28 12 FE 30
8780 38 13 FE 3A 30 0F CD 4B 86 18 0D CD 6E 86 18 08
8790 13 1A 5F 18 03 CD 9C 86 73 23 C9 01 00 00 2A 4A
87A0 FF ED 5B 48 FF 7E B7 CA F3 85 1A BE 20 08 7E FE
87B0 0D C8 23 13 18 F4 7E FE 0D 28 0A 23 7E FE 0D 20
87C0 FA 23 03 18 DC 1A FE 20 C8 FE 2B C8 FE 2D C8 18
87D0 F0 60 69 29 ED 4B 4C FF 09 5E 23 56 C9 13 ED 53
87E0 48 FF C5 CD 4B 86 C1 05 CD D1 87 62 6B D1 19 EB
87F0 C9 13 ED 53 48 FF C5 CD 4B 86 7B 2F 5F 7A 2F 57
8800 13 18 E3 18 17 E5 CD 9B 87 1A FE 2B 28 09 FE 2D
8810 28 0A CD D1 87 E1 C9 CD DD 87 E1 C9 CD F1 87 E1
8820 C9 CD F3 86 1A FE 24 28 10 FE 30 38 11 FE 3A 30
8830 0D CD 4B 86 73 23 72 23 C9 CD 6E 86 18 F6 CD 05
8840 88 18 F1 79 07 07 07 77 23 CD F3 86 3A 50 FF 3C
8850 28 26 CD 05 88 E5 ED 4B 56 FF B7 EB ED 42 EB E1
8860 E5 EB B7 ED 52 2B 7C B7 28 07 FE FF 28 0C C3 14
8870 86 7D FE 80 30 F8 E1 77 23 C9 7D FE 80 38 EF 18
8880 F5 ED 5B 52 FF E5 ED 4B 56 FF 09 7D 12 13 7C 12
8890 13 ED 53 52 FF E1 C9 CD F3 86 CD 59 87 79 FE 04
88A0 D2 E1 85 07 07 07 07 80 77 23 C9 CD F3 86 CD 5F
88B0 87 18 EA 36 CD 23 73 23 72 23 C9 CD DD 86 1A FE
88C0 0D 28 1A CD E1 88 CD DD 86 CD 47 8E CD 28 8E 3A
88D0 5C FF B7 28 E6 CD 3A 93 CD 44 81 18 DE CD 28 8E
```

```
88E0 C9 CD C0 86 22 58 FF ED 53 5A FF 1A FE 21 20 09
88F0 13 CD 7C 81 D8 77 23 18 F8 FE 3B 20 02 D1 C9 FE
8900 23 20 0B 13 ED 53 48 FF CD 4B 86 73 23 C9 FE 3D
8910 20 0E E5 2A 52 FF 2B 2B CD 21 88 CD 9D 8E E1 C9
8920 E5 CD F8 86 E1 79 FE 68 20 42 1A FE 52 20 16 13
8930 ED 53 48 FF CD 65 87 79 FE 08 CA E9 85 07 07 07
8940 C6 C0 77 23 C9 FE 4A 20 19 06 C2 13 ED 53 48 FF
8950 CD 65 87 79 FE 08 CA E9 85 07 07 07 80 77 23 C3
8960 21 88 FE 43 C2 E9 85 06 C4 13 18 E0 FE 08 30 0B
8970 07 07 07 FE 10 D2 43 88 77 23 C9 FE 09 28 0A 30
8980 0D 06 01 CD 97 88 C3 21 88 06 09 C3 97 88 FE 0B
8990 28 18 30 1A 06 02 CD F3 86 CD 59 87 79 FE 02 D2
89A0 E1 85 07 07 07 07 80 77 23 C9 06 0A 18 E8 FE 10
89B0 28 0E 30 11 D6 0C 07 07 07 C6 22 77 23 C3 21 88
89C0 06 03 C3 97 88 FE 12 28 07 30 1A 06 0B C3 97 88
89D0 06 04 CD F3 86 CD 40 87 79 FE 08 CA E1 85 07 07
89E0 07 80 77 23 C9 FE 14 28 06 30 15 06 05 18 E3 CD
89F0 F3 86 CD 40 87 79 FE 08 CA E1 85 06 06 C3 6B 87
8A00 FE 1D 28 0C 30 3A D6 15 07 07 07 C6 07 77 23 C9
8A10 CD F3 86 CD 40 87 79 FE 08 CA E1 85 47 ED 5B 48
8A20 FF 13 1A FE 2C C2 E1 85 13 ED 53 48 FF CD 40 87
8A30 79 FE 08 CA E1 85 78 07 07 07 81 C6 40 77 23 C9
8A40 FE 26 28 1A 30 10 D6 1E 07 07 07 47 CD F3 86 CD
8A50 40 87 79 FE 08 CA E1 85 80 C6 80 77 23 C9 06 C1
8A60 C3 AB 88 FE 27 20 05 06 C5 C3 AB 88 FE 30 28 0A
8A70 30 20 D6 28 4F 06 C6 C3 6B 87 CD F3 86 1A FE 30
8A80 DA E1 85 FE 3A D2 E1 85 D6 30 07 07 07 C6 C7 77
8A90 23 C9 FE 32 28 08 30 0A 36 C3 23 C3 21 88 36 CD
8AA0 18 F8 FE 34 28 06 30 0A 36 C9 23 C9 36 D3 23 C3
8AB0 72 87 FE 35 20 04 36 DB 18 F4 FE 3A 28 0C 30 0E
8AC0 D6 36 07 07 07 C6 E3 77 23 C9 36 D9 23 C9 FE 3C
8AD0 28 06 30 08 36 E9 23 C9 36 F9 23 C9 FE 3E 28 06
8AE0 30 08 36 DD 23 C9 36 FD 23 C9 FE 40 28 0A 30 0C
8AF0 06 42 36 ED 23 C3 97 88 06 4A 18 F6 FE 42 28 0D
8B00 30 0F 06 43 36 ED 23 CD 97 88 C3 21 88 06 4B 18
8B10 F3 FE 46 28 3E 30 43 D6 42 0F 0F 32 61 10 36 CB
8B20 23 CD F3 86 1A FE 30 DA E1 85 FE 3A D2 E1 85 D6
8B30 30 07 07 07 47 13 1A FE 2C C2 E1 85 13 ED 53 48
8B40 FF CD 40 87 79 FE 08 CA E1 85 80 47 3A 61 10 80
8B50 77 23 C9 36 ED 23 36 44 23 C9 FE 47 20 04 36 76
8B60 23 C9 FE 4C 30 0E 06 A1 D6 48 07 07 07 80 36 ED
8B70 23 77 23 C9 FE 50 28 08 30 19 06 A0 D6 4C 18 EA
8B80 E5 26 00 CD 21 88 EB ED 5B 54 FF B7 ED 52 22 56
8B90 FF E1 C9 FE 52 CA 21 88 30 1C CD F3 86 1A FE 22
8BA0 C2 E1 85 13 1A FE 0D 28 08 FE 22 28 04 77 23 18
8BB0 F2 ED 53 48 FF C9 FE 53 20 28 CD F3 86 CD 59 87
8BC0 79 FE 03 D2 E1 85 07 07 07 07 47 CD F3 86 CD 59
8BD0 87 79 FE 03 D2 E1 85 87 80 C6 40 77 23 C6 09 77
8BE0 23 C9 FE 55 28 1E 30 36 CD F3 86 CD 59 87 79 FE
8BF0 03 D2 E1 85 87 36 CB 23 C6 21 77 23 36 CB D6 11
8C00 23 77 23 C9 CD F3 86 CD 59 87 79 FE 03 D2 E1 85
8C10 87 36 CB 23 C6 03 77 23 36 CB D6 1F 23 77 23 C9
8C20 FE 57 28 15 30 26 36 1A 23 36 6F 23 36 13 23 36
8C30 1A 23 36 67 23 36 1B 23 C9 36 7D 23 36 12 23 36
8C40 13 23 36 7C 23 36 12 23 36 1B 23 C9 FE 59 28 4D
8C50 30 51 CD F3 86 1A FE 22 C2 E1 85 13 36 C3 23 E5
8C60 23 23 1A FE 0D 28 09 FE 22 28 05 77 23 13 18 F2
8C70 ED 53 48 FF 36 00 23 36 21 ED 4B 56 FF 54 5D 09
8C80 44 4D E1 71 23 70 23 42 4B ED 5B 56 FF 19 E5 C5
8C90 E1 C1 23 71 23 70 23 11 ED 52 C3 B3 88 11 75 0F
8CA0 C3 B3 88 FE 5B 28 0E 30 12 36 3E 23 CD 72 87 11
8CB0 A6 40 C3 B3 88 11 AC 0F C3 B3 88 FE 5D 28 08 30
8CC0 12 11 8A 1B C3 B3 88 36 3E 23 36 00 23 11 A6 40
8CD0 C3 B3 88 FE 5F 28 0E 30 12 36 3E 23 36 0A 23 11
8CE0 A6 40 C3 B3 88 11 7C 81 C3 B3 88 FE 61 28 08 30
8CF0 0C 11 92 81 C3 B3 88 11 3B 91 C3 B3 88 FE 63 28
8D00 06 30 1A 11 36 91 C3 B3 88 CD ED 86 79 FE 08 CA
8D10 E1 85 07 07 07 C6 04 77 23 3C 77 23 C9 FE 65 28
8D20 18 30 2D CD F3 86 CD 59 87 79 FE 03 D2 E1 85 87
8D30 C6 78 77 23 C6 39 77 23 C9 CD F3 86 CD 40 87 79
8D40 FE 08 CA E1 85 07 07 07 C6 06 77 23 36 00 23 C9
8D50 FE 67 30 0B 06 01 CD 97 88 AF 77 23 77 23 C9 26
8D60 00 CD 21 88 EB 22 54 FF 22 58 FF C9 2A 42 FF 22
8D70 40 FF 11 D6 8D CD 53 81 CD 06 81 11 96 8D CD 92
8D80 81 DA C3 8F 22 4A FF 11 E8 8D CD 53 81 EB 2A 40
8D90 FF 7E FE 3B 28 27 FE 0D 28 23 FE 20 28 1F B7 28
8DA0 2A FE 2B CA 04 86 FE 2D CA 04 86 FE 20 28 0A FE
8DB0 0D 28 06 12 13 23 7E 18 E8 3E 00 12 13 2A 40 FF
8DC0 7E 23 FE 0D 20 FA 22 40 FF 18 C3 12 13 ED 53 4C
8DD0 FF ED 53 52 FF C9 4C 41 42 45 4C 20 4E 41 40 45
8DE0 20 46 52 4F 4D 20 3F 00 50 41 53 53 20 30 0D 3A
8DF0 5C FF B7 C8 E5 2A 48 FF 7E F5 ED 5B 5A FF 36 0D
8E00 CD 53 81 F1 77 22 5A FF E1 C9 3E 10 32 64 EA 18
8E10 DE 3E 1C CD 0C 8E C9 ED 5B 48 FF D5 CD F3 86 0D
8E20 0A 8E D1 ED 53 48 FF C9 3E 17 32 64 EA ED 5B 48
8E30 FF D5 ED 5B 5A FF ED 53 48 FF CD DD 86 CD EF 8D
8E40 D1 ED 53 48 FF 18 CA 3A 5C FF B7 C8 3E 01 32 64
8E50 EA E5 2A 58 FF ED 5B 56 FF 19 CD 36 91 CD 4E 81
8E60 E1 E5 ED 5B 58 FF B7 ED 52 EB 16 05 7B B7 28 2B
8E70 15 CC 7C 8E 1D 7E CD 3B 91 23 18 F0 D5 E5 D5 E5
8E80 CD 28 8E E1 D1 CD 3A 93 CD 44 81 ED 5B 56 FF 19
8E90 CD 36 91 CD 4E 81 E1 D1 16 04 C9 E1 C9 3E 07 32
8EA0 64 EA 3A 5C FF B7 C8 5B CD 36 91 EB C9 2A 42 FF
8EB0 22 40 FF 22 48 FF 22 5A FF 2A 4C FF 22 52 FF 11
8EC0 00 00 EB 22 56 FF ED 5B 40 FF ED 53 48 FF ED 53
8ED0 5A FF 1A FE 3B 28 31 FE 0D 28 5B FE 20 28 24 B7
8EE0 C8 CD 81 88 CD 17 8E CD DD 86 1A FE 00 20 14 3A
8EF0 5C FF B7 28 0E 22 58 FF CD 47 8E CD 3A 93 CD 44
8F00 81 18 3D CD BB 88 18 2B CD B1 86 ED 53 48 FF 3E
8F10 01 32 64 EA CD EF 8D 3A 5C FF B7 28 03 CD 3A 93
8F20 CD 44 81 CD F1 0C C2 38 8F CD AC 0F FE 20 CC 75
8F30 0F 18 93 CD B1 86 18 EB CD 44 81 C3 C3 8F CD 6C
8F40 8D 3E FF 32 5D FF 3C 32 5C FF CD 44 81 11 8F 8F
8F50 CD 53 81 CD 44 81 CD AD 8E CD 44 81 2A 52 FF 23
8F60 36 00 7E B7 20 1D 3E 01 32 5D FF 3A 5D FF 32 5C
8F70 FF CD 44 81 11 96 8F CD 53 81 CD 44 81 CD AD 8E
8F80 C3 44 81 11 9D 8F CD 44 81 CD 53 81 C3 C3 8F 50
8F90 41 53 53 20 31 00 50 41 53 53 20 32 00 4C 41 42
8FA0 45 4C 20 54 41 42 4C 45 20 46 55 4C 4C 00 11 86
8FB0 90 CD 53 81 21 00 94 AF 32 5E FF 22 42 FF 36 00
8FC0 22 40 FF 31 FF FF CD 44 81 3E 2A CD A6 40 CD 06
8FD0 81 1A 13 FE 45 28 2C FE 50 CA 15 91 FE 41 28 28
8FE0 FE 42 28 20 FE 23 CA 21 90 FE 4A 28 2E FE 4D 28
8FF0 6C FE 4F CA A7 90 FE 21 CA FD 8F 18 CC CD 2C 5C
9000 C3 C3 8F CD 60 85 18 BB AF 32 5D FF CD 3E 8F 18
9010 B2 3E 01 32 5D FF CD 3E 8F 18 A8 CD 92 81 38 A3
9020 E9 CD 27 90 C3 C3 8F 3A 5E FF B7 28 09 AF 32 5E
9030 FF 11 4F 90 18 07 3C 32 5E FF 11 43 90 CD 44 81
9040 C3 53 81 50 52 49 4E 54 45 52 20 53 45 54 00 50
9050 52 49 4E 54 45 52 20 52 45 53 45 54 00 CD 92 81
9060 CD 44 81 CD 36 91 06 08 CD 4E 81 7E CD 3B 91 23
9070 10 F6 CD AC 0F B7 C4 75 0F CD AC 0F FE 58 CA C3
9080 8F CD 3A 93 18 DA 0C 2A 2A 20 50 43 2D 38 30 30
9090 31 20 41 53 53 45 4D 42 4C 45 52 20 56 45 52 20
90A0 35 2E 30 20 2A 2A 00 11 00 91 CD 53 81 CD 3A 93
90B0 CD 44 81 ED 4B 4A FF ED 5B 4C FF CD EF 90 D9 CD
90C0 4E 81 11 11 91 CD 53 81 D9 1A 6F 13 1A 67 1B C5
90D0 CD 36 91 C1 CD 3A 93 CD AC 0F B7 C4 75 0F CD AC
90E0 0F FE 78 CA C3 8F 0A B7 CA C3 8F 13 13 18 CC D5
90F0 50 59 CD 44 81 CD 53 81 0A 03 FE 00 20 FA D1 C9
9100 2A 20 4C 41 42 45 4C 20 20 54 41 42 4C 45 20 2A
9110 00 3D 20 24 00 CD 06 81 3E 01 32 49 EB 1A CD 18
9120 00 1A FE 0D CA 2A 91 13 18 F3 3E 0A CD 18 00 AF
9130 32 49 EB C3 C3 8F 7C CD 3B 91 7D F5 E6 F0 0F 0F
9140 0F 0F CD 48 91 F1 E6 0F B7 27 C6 F0 CE 40 C3 A6
9150 40 4E 4F 50 00 45 58 00 44 4A 4E 5A 00 4A 52 00
9160 4A 52 4E 5A 00 4A 52 5A 00 4A 52 4E 43 00 4A 52
9170 43 00 4C 58 49 00 44 41 44 00 53 54 41 58 00 4C
9180 44 41 58 00 53 48 4C 44 00 4C 48 4C 44 00 53 54
9190 41 00 4C 44 41 00 49 4E 58 00 44 43 58 00 49 4E
91A0 52 00 44 43 52 00 4D 56 49 00 52 4C 43 00 52 52
91B0 43 00 52 41 4C 00 52 41 52 00 44 41 41 00 43 4D
91C0 41 00 53 54 43 00 43 4D 43 00 4D 4F 56 00 41 44
91D0 44 00 41 44 43 00 53 55 42 00 53 42 42 00 41 4E
91E0 41 00 58 52 41 00 4F 52 41 00 43 4D 50 00 50 4F
91F0 50 00 50 55 53 48 00 41 44 49 00 41 43 49 00 53
9200 55 49 00 53 42 49 00 41 4E 49 00 58 52 49 00 4F
9210 52 49 00 43 50 49 00 52 53 54 00 4A 4D 50 00 43
9220 41 4C 4C 00 52 45 54 00 4F 55 54 00 49 4E 00 58
9230 54 48 4C 00 58 43 48 47 00 44 49 00 45 49 00 45
9240 58 58 00 50 43 48 4C 00 53 50 48 4C 00 49 58 00
9250 49 59 00 53 42 43 00 41 44 58 00 53 58 44 00 4C
9260 58 44 00 42 49 54 00 52 45 53 00 53 45 54 00 4E
9270 45 47 00 48 4C 54 00 43 50 49 4E 00 43 50 44 0C
9280 43 50 49 52 00 43 50 44 52 00 4C 44 49 00 4C 44
9290 44 00 4C 44 49 52 00 4C 44 44 52 00 4F 52 47 00
92A0 4D 53 47 00 44 57 00 4D 56 58 00 53 4C 58 00 53
92B0 52 58 00 4C 44 58 00 53 54 58 00 50 52 49 4E 54
92C0 00 47 45 54 4B 59 00 50 55 54 00 49 4E 4B 45 59
92D0 00 47 45 54 4C 4E 00 43 52 00 4C 46 00 4C 41 41
92E0 00 4C 41 48 00 50 52 41 00 50 52 48 00 54 53 54
92F0 00 54 53 58 00 43 4D 52 0C 43 4D 58 00 4F 46 53
9300 45 54 00 42 00 43 00 44 00 45 00 48 00 4C 00 4D
9310 00 41 00 42 00 44 0C 48 0C 50 53 57 00 42 00 44
9320 00 48 00 53 50 00 45 5A 00 5A 00 4E 43 00 43 00
9330 50 4F 00 50 45 00 50 00 4D 00 3A 5E FF B7 C8 E5
9340 D5 C5 21 01 01 22 94 ED 7D 32 96 ED CD A6 1E 2D
9350 28 05 3E 01 CD 86 0B 3E 00 CD A2 1B 11 96 EC 3E
9360 01 32 49 EB 1A B7 28 06 CD 18 00 13 18 F6 3E 0A
9370 CD 18 00 AF 32 49 EB C1 D1 E1 C9 FF 00 FF 00 FF
```



PC-8001

BASICによる逆アセンブラ

椚田鷹男

PC-8001のN-BASICは使いやすいのですが，マシン語レベルで何かやろうと思ってもモニタ・プログラムがまったく公表されていないので，手も足も出ません．

事務処理などはBASICだけで充分ですが，我々アマチュアとしては，やはりマシン語も使ってみたいものです．

そこで，まずモニタ・プログラムが見てみたくなり，逆アセンブラをBASICで使ってみました．

〔1〕プログラムの説明

実行速度を少しでも速くしようと思って，サブルーチンを前の方に持ってきてあります．そのため，プログラムがやや読みにくくなっているかもしれません．

あとでわかったことですが，サブルーチンをどこに置いても実行速度にはまったく関係なかったのです(後述)．参考までに，逆アセンブルする平均速度は3.5行/秒でした．

以下，プログラムの内容を簡単に説明します．

行番号	処理
50	B$にニモニックを付け加える．
100～ 110	1バイト読み取りB$に付け加える．
200～ 295	B$にオペランドを付け加える．(テーブル使用)
300	OPコードにCBHがある場合の処理．
400	オペランドに規則性がある場合の処理．
600	該当するニモニックがない場合の処理．
1000～1030	1バイト目が40H～BFHの場合の処理．
2000～2010	1バイト目がCBHの場合の処理．
3000～3080	1バイト目がDDHかFDHの場合の処理．(IXかIYが必ずオペランドにある)
4000	B$の初期設定．
4010	1バイト目を判定して，1000～3000の条件と合えばそこへ飛ばす．
4020～4060	1バイト目が1000～3000の条件以外のものの処理．
5000～5030	まとめて逆アセンブルするときに飛んで来るところ．
6000～6040	少しずつ逆アセンブルするときに飛んで来るところ．
7000～7080	配列の初期値設定．5000か6000へ飛ばす．

〔2〕変数の説明

1．配列変数について

配列の中身はどんなものかというと，NM$には8090～8100にあるニモニックが入り，OP$には9110～9120にあるオペランドが入ります．また，T，TD，IDにはそれぞれ8000～8040，8130～8140，8150～8160のDATAが数値として入ります．

この6桁の数が何を意味するのかというと，下位2桁がニモニックを，中位2桁が第1オペランドを，上位2桁がカンマに続く第2オペランドをそれぞれ表わします．

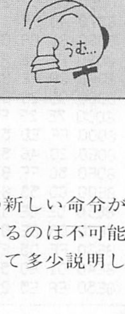
たとえば，000137ならばニモニックはNM$の37番目に入っている“INC”であり，第1オペランドはOP$の1番目に入っている“B”であり，上位2桁が00であるので第2オペランドはないことがわかり，結局，000137は“INC B”を表わしているのです．

もちろん，OPコードが696個もあるZ80の命令全部をこの方法で表わしているわけではなく，規則性に乏しい部分についてのみ，この方法を用いました．

2．単純変数について

変数名は何文字であってもかまいませんが，有効なのは最初の2文字だけです．以下，主な単純変数の意味について述べます．

SA：逆アセンブル開始アドレス（1バイト読むごとにインクリメントされていく）
EA：逆アセンブル終了アドレス
CL：行数のカウント
DT：PEEKしたデータ
T1：配列Tの要素のうちの1つ
T2：T1のうちの2桁
B$：1行分の逆アセンブル・バッファ



〔3〕新しい命令

このPC-8001のN-BASICにはいくつかの新しい命令が見受けられます．ここでそのすべてを紹介するのは不可能なので，本プログラムに使用した命令について多少説明します．

1．DEFINT 〈文字範囲〉……整　数

SNG 〈文字範囲〉……単精度

DBL 〈文字範囲〉……倍精度

STR 〈文字範囲〉……文　字



文字範囲内の文字で始まる変数を各々の型に宣言する(FORTRANでいう型宣言文である)．

ただし，プログラム中において個々に%(整数)，！(単精度)，#(倍精度)，$(文字)を付加した場合は，型宣言に優先します．また，型宣言で重複した場合は，最後に実行した型宣言が優先します．

2．MID$

a) MID$(p1$, p2, p3)　p2，p3：0～255

文字列p1$の始めからp2番目の文字よりp3個の文字列をこの関数値として与えます．

b) MID$(p1$, p2, p3)＝p4$

文字列p1中の任意に指定された(p2にて)位置よりp3個の文字列に文字列p4の頭からp3個の文字列をp1中に挿入します．

3．INSTR(p1, p2$, p3$) p1：0～255

文字列p2$中の最初にp3$の文字列が現われる位置を走査し，何番目にあるかを関数値として与えます．p1で走査を始める位置を指定しますが，特に指定がなければp1＝1として走査します．もちろん，走査する文字列がなければ関数値は0になります．

〔4〕使用方法

まず，**リスト1**のプログラムを入力してください．DATA文中の数字や文字は見やすいように並べてありますが，余分なブランクや数字の頭に付いている0は省略した方がメモリ効率は良くなります．

RUNすると，少しずつ逆アセンブルしたいかどうか聞いてきます．CRTを見ながら少しずつプログラムを追っていくような場合は，N以外の文字を入力してください．すると，逆アセンブル開始番地とそこから何行逆アセンブルするか聞いてくるので，アドレスは16進数で，行数は10進数で入力します．その場合，マイナス記号で2つの数字を区切ってください．

なお，アドレスおよび行数はどちらも省略可能で，アドレスを省略すれば前の続きを逆アセンブルしますし，行数を省略すれば18行と見なします．

次に，まとめて逆アセンブルしたいときはRUNした後でNを入力して，逆アセンブル開始番地と終了番地を16進数で入力します．

また，逆アセンブルしたものをプリンタに出力したい場合には，行番号5030と6040にあるPRINT文をLPRINT文に変更してください．実行例を**リスト2**に示します．

〔5〕逆アセンブラによる収穫

PC-8001のモニタ・プログラムを逆アセンブルしてみたところ，使えそうなサブルーチンがいくつか見つかったので，**表1**に示します．

この中で，Flow AccというのはF0A8H～F0ABHの4バイトのことで，演算する場合にはここをよく使います．Reg Pairというのは，BCDEの4つのレジスタをペアにしたもののことです．単精度演算の場合，F0ABHおよびBレジスタに指数部が入り，以下，順に仮数部の上位・中位・下位が入ります．

モニタ・プログラムを読む場合に，プログラム領域かデータ領域かがわからず，苦労したことがありました．そこで，これからモニタ・プログラムを読もうとする人のために，主なデータ領域を**表2**に示しましたので参考にしてください．



▲システム風景

表1　モニタ・プログラム内のサブルーチン

アドレス	用　途	使い方・動作
0F75	1文字入力	キーボードが押されるまで入力待ちとなる．結果はAccにASCIIコードで入る．
0FAC	1文字入力	キーボードが押されるまで待たない．結果はAccにASCIIコードで入る．
1B8A	1行入力	EC96Hからバッファ(最後の文字の次には0が入る)．
240F	単精度減算	(Flow Acc)←(Reg Pair)－(Flow Acc)
2412	単精度加算	(Flow Acc)←(Reg Pair)＋(Flow Acc)
2541	単精度乗算	(Flow Acc)←(Reg Pair)×(Flow Acc)
259C	単精度除算	(Flow Acc)←(Reg Pair)÷(Flow Acc)
309F	HEX→DEC	(F0A8)～(F0A9)の16進数を5桁の10進数の文字列にして(HL)～(HL＋4)に入れ，(HL＋5)には0を入れる．
3BF9	エラー表示	EレジスタにERRコードを入れておく(必ずBASICモードのコマンド待ちになる)．
40A6	1文字出力	AccにASCIIコードを入れていく．
44C7	DEC→HEX	HLに文字列の先頭アドレスを入れていくと，それを16進数に変換したものがDEレジスタに入る．
52ED	文字列の出力	HLに文字列の先頭アドレスを入れていくと，0があるまで出力する．

表2　モニタ・プログラム内のDATA領域

アドレス	DATAの使用目的
10E5～117C	キーボード・マトリクスをASCIIコードに変換する．
3148～3151	16進数を10進数に変換する．
33BD～3466	中間言語に対応した処理ルーチンをCALLする．
3467～34C0	関数の中間言語に対応した処理ルーチンをCALLする．
34C1～37A7	コマンドを中間言語に変換する．
37A8～37BC	特殊記号を中間言語に変換する．
37F2～3A04	エラーメッセージのテーブル
3A05～3B5B	初期設定のとき，EA00H～EB56Hに転送する．
5C6F～5C92	モニタ・モードにおいて，コマンドに対応した処理ルーチンをCALLする．

リスト1　逆アセンブラ・リスト

```
0   GOTO 7000
50  MID$(B$,22,LEN(NM$(T2)))=NM$(T2):
    RETURN
100 DT=PEEK(SA): W$=HEX$(DT):
    IF LEN(W$)=1 THEN W$="0"+W$
110 MID$(B$,DP,2)=W$: SA=SA+1: DP=DP+3:
    RETURN
200 IF T2<52 THEN N=0 ELSE N=1: T2=T2-50
210 TT$=OP$(T2): P=INSTR(TT$," "):
    IF P>0 THEN N=N+1
220 ON N+1 GOTO 230,240,270
230 B$=B$+TT$: RETURN
240 GOSUB 100: IF OT2<42 OR OT2>44 THEN
    W1$=W$: GOTO 280
250 W1$="0000": TA=VAL("&H"+W$+"00")/256
    +SB+1
260 W=LEN(HEX$(TA)): MID$(W1$,5-W,W)=
    HEX$(TA): GOTO 280
270 GOSUB 100: W1$=W$: GOSUB 100:
    W1$=W$+W1$
280 IF P>1 THEN B$=B$+LEFT$(TT$,P-1)
290 B$=B$+W1$+"H": IF LEN(TT$)>P THEN
    B$=B$+RIGHT$(TT$,LEN(TT$)-P)
295 RETURN
300 T2=DT¥&H40+16: GOSUB 50:
    T2=(DT MOD &H40)¥8+16:
    B$=B$+OP$(T2)+",": RETURN
400 T2=(DT MOD 8)+1: B$=B$+OP$(T2):
    RETURN
500 N=T1/100: T2=T1-N*100: T1=N: RETURN
600 SA=SB: B$=LEFT$(B$,10)+SPACE$(11)+
    "???": RETURN
1000 IF DT>&H7F THEN T2=DT¥8-7: GOSUB50:
     IF T2>12 OR T2=11 THEN 400 ELSE
     T2=8: GOTO 1030
1010 IF DT=&H76 THEN T2=34: GOTO 50
1020 T2=20: GOSUB 50: T2=DT¥8-7
1030 B$=B$+OP$(T2)+",": GOTO 400
2000 GOSUB 100: IF DT<&H40 THEN
     T2=DT¥8+1: IF T2=7 THEN 600 ELSE
     GOSUB 50: GOTO 400
2010 GOSUB 300: GOTO 400
3000 IF DT=&HFD THEN Y=1 ELSE Y=0
3010 GOSUB 100: IF DT=&HCB THEN 3070
     ELSE IF DT<ID(20) THEN 3040
3020 FOR I=20 TO 38: IF DT=ID(I) THEN
     3060 ELSE IF DT>ID(I) THEN NEXT
3030 GOTO 600
3040 FOR I=19 TO 1 STEP -1: IF DT=ID(I)
     THEN 3060 ELSE IF DT<ID(I)
     THEN NEXT
3050 GOTO 600
3060 T1=TD(I)+Y: GOTO 4040
3070 GOSUB 100: W1$=W$: GOSUB 100:
     IF DT MOD 8 <>6 THEN 600
     ELSE GOSUB 300
3080 B$=B$+"(I"+CHR$(&H58+Y)+"+"+W1$+
     "H)": RETURN
4000 B$="0000-"+SPACE$(21):
     W=LEN(HEX$(SA)): MID$(B$,5-W,W)=
     HEX$(SA): DP=8
4010 GOSUB 100: SB=SA: IF DT>&H3F AND
     DT<&HC0 THEN 1000 ELSE IF DT=&HCB
     THEN 2000 ELSE IF DT=&HDD OR
     DT=&HFD THEN 3000
4020 IF DT=&HED THEN GOSUB 100:
     IF DT<&H40 OR DT>&HBF THEN 600
4030 T1=T(DT+1): IF T1=0 THEN 600
4040 GOSUB 500: GOSUB 50: OT2=T2
4050 IF T1=0 THEN RETURN ELSE GOSUB 500:
     GOSUB 200
4060 IF T1=0 THEN RETURN ELSE B$=B$+",":
     T2=T1: GOTO 200
5000 INPUT "START-ADRRESS";SA$
5010 INPUT "E N D-ADRRESS";EA$:
     IF LEN(EA$)=0 THEN 5010 ELSE
     IF LEN(SA$)=0 THEN SA=0 ELSE
     SA=VAL("&H"+SA$): IF SA<0 THEN
     SA=SA+65536!
5020 EA=VAL("&H"+EA$): IF EA<0 THEN
     EA=EA+65536!
5030 IF SA>EA THEN END ELSE GOSUB 4000:
     PRINT B$: GOTO 5030
6000 SA=0
6010 CL=18: SA$="": PRINT: INPUT
     "ADRRESS-LINE";SA$: IF LEN(SA$)=0
     THEN 6040 ELSE N=INSTR(SA$,"-"):
     IF N=0 THEN N=LEN(SA$)+1
6020 IF N>1 THEN SA=VAL("&H"+LEFT$(SA$,
     N-1)): IF SA<0 THEN SA=SA+65536!
6030 IF N<LEN(SA$) THEN CL=VAL(RIGHT$(
     SA$,LEN(SA$)-N))
6040 GOSUB 4000: PRINT B$: CL=CL-1:
     IF CL>0 THEN 6040 ELSE 6010
7000 DEFINT A-Z: DEFSNG E,S,T
7010 DIM T(256),NM$(67),OP$(51),TD(39),
     ID(39)
7020 FOR N=  1 TO 124: READ   T(N):NEXT
7030 FOR N=161 TO 256: READ   T(N):NEXT
7040 FOR N=  1 TO  67: READ NM$(N):NEXT
7050 FOR N=  1 TO  51: READ OP$(N):NEXT
7060 FOR N=  1 TO  39: READ  TD(N):NEXT
7070 FOR N=  1 TO  39: READ  ID(N):NEXT
7080 CONSOLE 0,25,0: INPUT"STEP BY STEP"
     ;SA$: IF LEN(SA$)=0 THEN 6000 ELSE
     IF LEFT$(SA$,1)="n" THEN 5000 ELSE
```

```
     6000
8000 DATA   000050,893320,084220,003337,
     000137,000129,390120,000062,383732,
     333509,420820,003329,000237,000229,
     390220,000065,003944,893420,084320,
     003437,000337,000329,390320,000061,
     003943,343509,430820,003429,000437,
     000429
8010 DATA   390420,000064,390943,893520,
     359020,003537,000537,000529,390520,
     000028,391043,353509,903520,003529,
     000637,000629,390620,000027,391143,
     893620,089020,003637,000737,000729,
     390720,000007,390243,363509,900820,
     003629
8020 DATA   000837,000829,390820,000022,
     410136,014153,333512,339020,000049,
     000060,001635,085120,410236,024153,
     333510,903320,000000,000059,000000,
     083220,410336,034153,343512,349020,
     000000,000000,001735,510820,410436,
     044153
8030 DATA   343510,903420,000000,000000,
     001835,320820,410536,054153,353512,
     000000,000000,000000,000000,000066,
     410636,064153,353510,000000,000000,
     000000,000000,000063,000000,000000,
     363512,369020,000000,000000,000000,
     000000
8040 DATA   410836,084153,363510,903620
8050 DATA 47,25,40,55,00,00,00,00,45,23,
      38,54,00,00,00,00,48,26,41,52,00,
      00,00,00,46,24,39,51,00,00,00,00
8060 DATA   000958,003356,890942,008942,
     890921,003357,390809,002467,001058,
     000058,891042,000000,891021,008921,
     390810,002567,001158,003456,891142,
     084053,891121,003457,003911,002667,
     000258,000033,890242,400836,890221,
     000000
8070 DATA   390812,002767,001258,003556,
     891242,354432,891221,003557,003913,
     002867,001358,000742,891342,353432,
     891321,000000,003914,002967,001458,
     003756,891442,000030,891421,003757,
     003915,003067,001558,353620,891542,
     000031
8080 DATA   891521,000000,003916,003167
8090 DATA RLC, RRC, RL,  RR,  SLA, SRA,
     SCF, SRF, ADD, ADC, SUB, SBC, AND,
     XOR, OR,  CP,  BIT, RES, SET, LD,
     CALL,CCF, CPD, CPDR,CPI, CPIR,CPL,
     DAA, DEC, DI,  EI,  EX,  EXX, HALT
8100 DATA IM,  IN,  INC, IND, INDR,INI,
     INIR,JP,  JR,  DJNZ,LDD, LDDR,LDI,
     LDIR,NEG, NOP, OTDR,OTIR,OUT, OUTD,
     OUTI,POP, PUSH,RET, RETI,RETN,RLA,
     RLCA,RLD, RRA, RRCA,RRD, RST
8110 DATA B,   C,   D,   E,   H,   L,
     (HL),A,   NZ,  Z,   NC,  PO,  PE,
     P,   M,   0,   1,   2,   3,   4,
     5,   6,   7,   00H, 08H, 10H, 18H,
     20H, 28H, 30H, 38H, R,   BC,  DE,
     HL,  SP,  AF,  AF'
8120 DATA " ", ( ), (C), (BC),(DE),(SP),
     (IX),(IY),(IX+ ),(IY+ ), IX, IY, I
8130 DATA   334909,344909,894920,499020,
     004937,494909,904920,004929,004737,
     004729,394720,364909,470120,470220,
     470320,470420,470520,470620,014720
8140 DATA   024720,034720,044720,054720,
     064720,084720,470820,470809,470810,
     004711,470812,004713,004714,004715,
     004716,004956,494432,004957,004542,
     493620
8150 DATA   009,025,033,034,035,041,042,
        043,052,053,054,057,070,078,086,
        094,102,110,112,113,114,115,116,
        117,119,126,134,142,150,158,166,
        174,182,190,225,227,229,233,249
```

《逆アセンブル実行例》

```
STEP BY STEP? y

ADRRESS-LINE? 62c-9
062C-   1E 13          LD    E,13H
062E-   7A             LD    A,D
062F-   01 02 00       LD    BC,0002H
0632-   09             ADD   HL,BC
0633-   BE             CP    (HL)
0634-   38 03          JR    C,0639H
0636-   1D             DEC   E
0637-   20 F9          JR    NZ,0632H
0639-   2B             DEC   HL
```

〔6〕PC-8001を使ってみて気付いたこと

- N-BASICは命令数も多く，たいていのことはできるのですが，どうしたことかPOP命令だけがありません．プログラムが読みにくくなるから使うな，ということでしょうか．
- N-BASICのマニュアルの99ページに，MAKINTルーチンのアドレスが24Hとなっていますが，これは23Hの間違いだと思われます．
- マシン語のプログラム中で無限ループに入ってしまった場合，当然**STOP**キーは役に立ちません．かといって，

リセットしてしまえばすべてがイニシャライズされてしまいます．そんなときは，**STOP** キーを押しながら **RESET** を押しましょう．画面はクリアされますが，イニシャライズはされません．もちろん，BASIC のプログラムや変数の値も残っています．

- 普通，ファンクション・キーには15文字までしか入りませんが，**POKE** 文かモニタ・モードの**Sコマンド**を用いればもっと入れられます．たとえば，**f･1**のバッファはEA7CH からですので，ここからASCII コードで入れていけば何文字でも（といっても，バッファ全体の大きさが160バイトなのでその範囲内で）入ります．最後に0を入れることも忘れないように！
- ファンクションキーが押されるとフラグがセットされ，ファンクション・キーのバッファのアドレスを示すポインタもセットされます．したがって，強制的にフラグとポインタを変えてやれば，入力待ちの状態になったときに，そのポインタからデータを持ってきます．

  フラグのアドレスはEA68Hで，ここに0以外のものが入っているとファンクション・キーが押されたものと見なされます．

  また，ポインタのアドレスはEDC0H～EDC1Hです．何かユニークな使い方を考えてみてはいかがでしょうか．
- **GOTO** 文など，行番号を必要とする命令を入力すると行番号は16進数に変換され，2バイトに収まります．ところが，RUN してその文が実行されると，そこに収まっていた行番号は，なんとアドレスに変わってしまうのです．つまり，その行番号のプログラムが実際に存在しているアドレスに変換されるわけです．そのために，この逆アセンブラ・プログラムのように，サブルーチンを前の方に持って来てもまるで意味がないことになります．もちろん，**LIST** すれば画面には行番号に変換されたものが出てきます．なお，リナンバーするときにも1度アドレスに変換してから行なっているようです．

〔7〕おわりに

モニタ・プログラムを読む場合に，文字列や数字の列を探し出すプログラムがあると，なにかと便利です．そこで，オマケとして search プログラムを**リスト3**に載せておきますので利用してください．また，N-BASIC の中間言語は**表3**のようになっています．表の中で，0DHと0EHの『アドレス』と『行番号』というのは，上で説明したもののことです．

PC-8001は使いやすいマイコンです．スクリーン・エディタやファンクション・キーなどの機能は便利です．



リスト3　SEARCH プログラム・リスト

```
10000 '
10001 '          << SEARCH >>
10002 '
10010 CLEAR 300,&HE7FF
10020 DEFINT A-Z:DIM MOJI%(10):GOSUB
       10280:DEFUSR0=&HE800
10030 CONSOLE,,1:PRINT "*** SEARCH ***"
10040 PRINT "STRING or HEX (S or H)? ";
10050 MOJI$=INKEY$:IF LEN(MOJI$)=0 THEN
       10050
10060 IF MOJI$="S" OR MOJI$="s" THEN
       PRINT "Str";:GOTO 10120
10070 IF MOJI$<>"H" AND MOJI$<>"h" THEN
       10080 ELSE PRINT "Hex";
10080 PRINT:INPUT "H E X DATA";MOJI$:
       N=LEN(MOJI$):IF N=0 THEN 10080
10090 I=1:J=1
10100 NB=INSTR(I,MOJI$," "):IF NB=0 THEN
       NB=LEN(MOJI$)+1
10110 MOJI(J)=VAL("&H"+MID$(MOJI$,I,
       NB-I)):I=NB+1:J=J+1:IF I<N THEN
       10100 ELSE N=J-1:GOTO 10140
10120 PRINT:INPUT "STRING"; MOJI$:N=LEN
       (MOJI$):IF N=0 THEN 10120
10130 FOR I=1 TO N:MOJI(I)=ASC(MID$(
       MOJI$,I,1)):NEXT
10140 STAD=   &H0:ADRS$="":
       INPUT "START ADRS";ADRS$:
       IF LEN(ADRS$)>0 THEN
       STAD=VAL("&H"+ADRS$)
10150 EDAD=&H5FFF:ADRS$="":
       INPUT "E N D ADRS";ADRS$:
       IF LEN(ADRS$)>0 THEN
       EDAD=VAL("&H"+ADRS$)
10160 IF EDAD=STAD THEN POKE &HE8FC,&HFF
       :GOTO 10220
10170 FOR I=1 TO N:POKE &HE8FF+I,MOJI(I)
       :NEXT
10180 POKE &HE8FE,N MOD 256:POKE &HE8FF,
       N¥256
10190 POKE &HE8FA,PEEK(VARPTR(STAD)):
       POKE &HE8FB,PEEK(VARPTR(STAD)+1)
10200 POKE &HE8F8,PEEK(VARPTR(EDAD)):
       POKE &HE8F9,PEEK(VARPTR(EDAD)+1)
10210 A=USR0(0)
10220 IF PEEK(&HE8FC)<>0 THEN PRINT
       "ERROR"
10230 BEEP:PRINT:GOTO 10030
10240 REM
10250 DATA ed,73,f6,e8,31,00,ea,f5,c5,
           d5,e5,af,2a,f8,e8,ed,4b,fa,
           e8,ed,42,38,12,44,4d,2a,fa,
           e8,11,00,e9,1a,ed,b1,20,0c
10260 DATA cd,40,e8,18,f3,3e,01,32,fc,
           e8,18,07,af,32,fd,e8,32,fc,
           e8,e1,d1,c1,f1,ed,7b,f6,e8,
           c9,f5,c5,d5,2b,e5,ed,4b,fe
10270 DATA e8,1a,be,20,17,0d,28,04,23,
           13,18,f5,cd,d4,5f,e1,e5,2b,
           cd,bd,5e,cd,d4,5f,23,cd,c0,
           5e,e1,d1,c1,f1,23,c9,00,00
10280 FOR I=&HE800 TO &HE869:READ J$:
       POKE I,VAL("&H"+J$):NEXT:RETURN
```

表3　中間言語対照表

TOKEN	KEYWORD
0 0	—
1	SH
2	SX
3	EX
4	ET
5	EQ
6	AK
7	BL
8	BS
9	HT
A	LF
B	&O
C	&H
D	アドレス
E	行番号
F	整数(10~255)
1 0	0 (1桁の整数のとき)
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	
B	
C	整数(256~)
D	単精度
E	1Fに同じ
F	倍精度
2 0	brank
1	!
2	"
3	#
4	$
5	%
6	&
7	'
8	(
9	)
A	*
B	+
C	,
D	—
E	.
F	/
3 0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	:
B	;
C	<
D	=
E	>
F	?

TOKEN	KEYWORD
4 0	@
1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H
9	I
A	J
B	K
C	L
D	M
E	N
F	O
5 0	P
1	Q
2	R
3	S
4	T
5	U
6	V
7	W
8	X
9	Y
A	Z
B	[
C	¥
D	]
E	^
F	_
6 0	
1	a
2	b
3	c
4	d
5	e
6	f
7	g
8	h
9	i
A	j
B	k
C	l
D	m
E	n
F	o
7 0	p
1	q
2	r
3	s
4	t
5	u
6	v
7	w
8	x
9	y
A	z
B	{
C	¦
D	}
E	~
F	

TOKEN	KEYWORD
8 0	—
1	END
2	FOR
3	NEXT
4	DATA
5	INPUT
6	DIM
7	READ
8	LET
9	GOTO
A	RUN
B	IF
C	RESTORE
D	GOSUB
E	RETURN
F	REM
9 0	STOP
1	PRINT
2	CLEAR
3	LIST
4	NEW
5	ON
6	WAIT
7	DEF
8	POKE
9	CONT
A	CSAVE
B	CLOAD
C	OUT
D	LPRINT
E	LLIST
F	CONSOLE
A 0	WIDTH
1	ELSE
2	TRON
3	TROFF
4	SWAP
5	ERASE
6	ERROR
7	RESUME
8	DELETE
9	AUTO
A	RENUM
B	DEFSTR
C	DEFINT
D	DEFSNG
E	DEFDBL
F	LINE
B 0	PRESET
1	PSET
2	BEEP
3	FORMAT
4	KEY
5	COLOR
6	TERM
7	MON
8	CMD
9	MOTOR
A	POLL
B	RBYTE
C	WBYTE
D	ISET
E	IRESET
F	TALK

TOKEN	KEYWORD
C 0	MAT
1	LISTEN
2	DSKO$
3	REMOVE
4	MOUNT
5	OPEN
6	FIELD
7	GET
8	PUT
9	SET
A	CLOSE
B	LOAD
C	MERGE
D	FILES
E	NAME
F	KILL
D 0	LSET
1	RSET
2	SAVE
3	LFILES
4	INIT
5	LOCATE
6	—
7	TO
8	THEN
9	TAB(
A	STEP
B	USR
C	FN
D	SPC(
E	NOT
F	ERL
E 0	ERR
1	STRING$
2	USING
3	INSTR
4	'
5	VARPTR
6	CSRLIN
7	ATTR$
8	DSKI$
9	INKEY$
A	TIME$
B	DATE$
C	
D	SRQ
E	STATUS
F	POINT
F 0	>
1	=
2	<
3	+
4	—
5	*
6	/
7	^
8	AND
9	OR
A	XOR
B	EQV
C	IMP
D	MOD
E	¥
F	function(*2)

function時の第2コード(*3)

TOKEN	KEYWORD
8 0	—
1	LEFT$
2	RIGHT$
3	MID$
4	SGN
5	INT
6	ABS
7	SQR
8	RND
9	SIN
A	LOG
B	EXP
C	COS
D	TAN
E	ATN
F	FRE
9 0	INP
1	POS
2	LEN
3	STR$
4	VAL
5	ASC
6	CHR$
7	PEEK
8	SPACE$
9	OCT$
A	HEX$
B	LPOS
C	PORT
D	DEC
E	BCD$
F	CINT
A 0	CSNG
1	CDBL
2	FIX
3	CVI
4	CVS
5	CVD
6	DSKF
7	EOF
8	LOC
9	LOF
A	FPOS
B	MKI$
C	MKS$
D	MKD$
E	—
F	—
E C	IEEE(*4)



＊NOTE 1：0Dのアドレスとは，RUNさせるとGOTOなどの行番号がその行のある先頭アドレスを示す．
0Eの行番号とは，RUN させる前のGOTO などの行番号を示す．
0Fの整数とは，以下1バイトは10進整数であることを示す．
1Cの整数とは，以下2バイトは10進整数であることを示す．
1Dの単精度とは，以下4バイトは単精度定数であることを示す．
1E, 1Fの倍精度とは，以下8バイトは倍精度定数であることを示す．

＊NOTE 2：FFのfunctionとは，FFに続く次のページの80~ADまでの中間言語のコードをもって，1つの関数を成す．

＊NOTE 3：この表にあるコードはこれだけでは意味がなく，このコードの前にFFのコードを付加して初めて関数になる．

＊NOTE 4：コードECについては何に使われているか不明．

＊この中間言語表によれば，ユーザーマニュアルには書かれていないSTATEMENTがあったりして，面白い．

PC-8001 BASICによる改良版逆アセンブラ

命令コードはザイログ仕様!!

佐藤　実
大垣泰二

'80年2月号に載っていた『PC-8001　BASICによる逆アセンブラ』を走らせてみたところ，いくつかの虫が見つかりました．全体としては良くできたプログラムなので，このまま眠らせておくのはもったいないと思い，手を入れて発表させてもらうことにしました．このプログラムは原作者の了解をとってカセット・サービスさせていただくことにしました．

1 逆アセンブラの最終テスト

最終テストはＰＣの＄Ｄ000～＄Ｄ587のエリアにＺ80の全命令696個がアルファベット順に並ぶように入力し，これを逆アセンブルして行ないました．これは手間がかかりますが，大変確実な方法です．カセットのＢ面におまけとして入れておくので，試してみてください．

このデータは，最初シャープのマシン・ランゲージ，SP-2001のマニュアルに載っていたものを入力したのですが，なんと279番目のＬＤA，Ｒと357番目のＬＤ　Ｒ，Ａが抜けていました．

また，ＩＮ Ａ，（Ｎ）とＯＵＴ（Ｎ），Ａのカッコも抜けているので，現在はシャープのシステム・プログラムＳＰ2101～2401に付いてくるプログラミング・マニュアルの後に載っているものを使っています．

ただし，オペランドの定数だけはマシン・ランゲージのマニュアルそのままを入れてあります．

2 de BUGと機能追加

まず，**実行例**を見てください．これは虫の出るコードのサンプルで，ａがオリジナル・プログラム，ｂが改良プログラムで逆アセンブルしたものです．

虫の出るコードを分類してみると，

❶ＤＣｎｎ➡ＣＡＬＬ　Ｃ，ｎｎ
❷先頭にＦＤが付く命令のすべて
❸ＤＤＦ９➡ＬＤ　ＳＰ，ＩＸとＦＤＦ９➡ＬＤ　ＳＰ，ＩＹ
❹ＤＤＣＢｄ06～ＤＤＣＢｄ３Ｅ➡ＲＬＣ　（ＩＸ＋ｄ）や，ＦＤＣＢｄ06～ＦＤＣＢｄ３Ｅ➡ＲＬＣ　（ＩＹ＋ｄ）など．

の４つに分けられます．この他にザイログのニモニックと異なるものを使っているのが２つありました．

これは行番号8090の文字データを変えればすぐ直ります（ＳＬＲ→ＳＬＡ，ＳＲＦ→ＳＲＬ）．

実行例

a）オリジナル・プログラムによるアセンブル例

```
D600-  DC 84 05     LD   C,0584H
D603-  FD 8E 05     SUB  A,(IX+05H)
D606-  FD 29        ADC  IX,IX
D608-  FD A6 05     XOR  (IX+05H)
D60B-  FD 2B        DI   IX
D60D-  FD 77 05     CALL (IX+05H),A
D610-  FD 7E 05     CALL A,(IX+05H)
D613-  DD           ???
D614-  F9           LD   SP,HL
D615-  FD           ???
D616-  F9           LD   SP,HL
D617-  DD CB 05 16  CP   2,(IX+05H)
D61B-  FD CB 05 16  CP   2,(IY+05H)
D61F-  DD CB 05 1E  CP   3,(IX+05H)
D623-  FD CB 05 1E  CP   3;(IY+05H)
```

b）改良版プログラムによる逆アセンブル例

```
D600-DC 84 05     CALL C,0584H     ワ■
D603-FD 8E 05     ADC  A,(IY+05H)  ■
D606-FD 29        ADD  IY,IY       )
D608-FD A6 05     AND  (IY+05H)    ヲ
D60B-FD 2B        DEC  IY          +
D60D-FD 77 05     LD   (IY+05H),A  w
D610-FD 7E 05     LD   A,(IY+05H)  ~
D613-DD F9        LD   SP,IX       ン
D615-FD F9        LD   SP,IY
D617-DD CB 05 16 RL    (IX+05H)    ンヒ
D61B-FD CB 05 16 RL    (IY+05H)    ヒ
D61F-DD CB 05 1E RR    (IX+05H)    ンヒ
D623-FD CB 05 1E RR    (IY+05H)    ヒ
```

プログラムはI/O'80年２月号のリストに，いま述べた４つのデバッグと３つの機能追加を行なっています．プログラム中，▒▒▒の部分が今回改良したところです．次に，具体的な変更点を述べます．

❶ＣＡＬＬ　Ｃ，ｎｎを表わす６桁のデータがない．
▶ＤＣコードをチェックして，見つけたＳＴ１に，正しい６桁のデータを入れています（**行番号4010**）．

❷Ｙフラグの取り扱いが間違っていて，オペランドのＩＸをＩＹに変えるべきところをオペコードの方を変えてしまっている．
▶間違っているところを削除して，新しくＹフラグの処理を追加します（**行番号3000，3060，3063，3064，3065**）．

❸39個あるテーブルのデータを38個までしか比較してくれないので，最後にあるＦ９コードを見逃してしまう．
▶38を39に変えるだけです（**行番号3020**）．

❹ＤＣＢＢ～，ＦＤＣＢ～の大部分はビット操作命令ですが，前の方の７個だけはローテート・シフト命令になっています．その処理がまったくされていません．
▶該当するコードをピックアップして処理します（**行番号3070，3071，3072**）．

その他の変更点は追加機能に関するもので，逆アセンブ

ラ本体には関係ありません．なくても逆アセンブル自体は正しく行なわれます．追加機能は，

a)プリンタへの出力とそのソフトウェア・スイッチのON/OFF(行番号7080，7085，5030，6040)．

b)逆アセンブル中のコードを画面右端にキャラクタで出力．ただし，20H以下のコードはすべてブランクになります(行番号50，110，600，4000，5030，6040)．

c)!キーの入力で逆アセンブルの中断，および再スタート(行番号5031，6010，6041)．

3 使い方

プログラムをスタートさせると，まず，ＬＰＲＩＮＴ？とプリンタ出力の有無を聞いてきます．y CR を入力すればプリンタ出力のソフトウェア・スイッチがＯＮ，その他の入力ではすべてＯＦＦになります．

次に，ＳＴＥＰ ＢＹ ＳＴＥＰ？と聞いてきます．これ以後はオリジナルと同じ手順[注)]ですが，プリンタ出力ＯＮのとき，『ステップ・バイ・ステップ・モード』では，プリンタと画面に同じものが逆アセンブル出力されますが，『連続モード』では画面には出力されません．

どちらのモードでも，!キー（シフトが必要）を押すことによってスタートへ戻ることができます．また,『連続モード』が終わったときもスタートへ戻ります．

B面に入っている『テスト用データ』はモニタでロードしてください．＄D000～＄D587にロードされているので，逆アセンブルして前記のプログラミング・マニュアルのリストと比べてみてください．このリストをすべて載せれば良いのでしょうが，なにしろかなりの量なのであしからず．

写真1　ステップ・バイ・ステップ時の逆アセンブル例

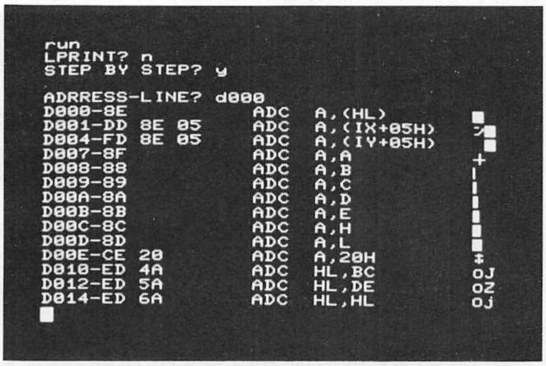
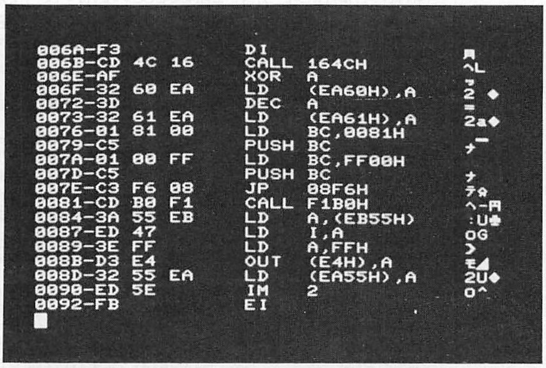


写真2　連続モードでの逆アセンブル例



◘参考引用文献

1)桝田鷹男："BASICによる逆アセンブラ",I/O，'80年2月号

改良版ＰＣ用逆アセンブラ プログラム・リスト

```
1 '* I/O 1980-02     *
2 '* ｷﾞｬｸ ｱｾﾝﾌﾞﾗ     *
3 '* 1980-12 Ver-2 *
10 GOTO 7000
50 MID$(B$,18,LEN(NM$(T2)))=NM$(T2):RETURN
100 DT=PEEK(SA): W$=HEX$(DT):IF LEN(W$)=1 THEN W$="0"+W$
110 MID$(B$,DP,2)=W$: SA=SA+1: DP=DP+3:IF DT<&H20 THEN BB$=BB$+" ":RETURN ELSE B
B$=BB$+CHR$(DT):RETURN
200 IF T2<52 THEN N=0 ELSE N=1:T2=T2-50
210 TT$=OP$(T2):P=INSTR(TT$," "):IF P>0 THEN N=N+1
220 ON N+1 GOTO 230,240,270
230 B$=B$+TT$:RETURN
240 GOSUB 100:IF OT2<42 OR OT2>44 THEN W1$=W$:GOTO 280
250 W1$="0000":TA=VAL("&H"+W$+"00")/256+SB+1
260 W=LEN(HEX$(TA)):MID$(W1$,5-W,W)=HEX$(TA):GOTO 280
270 GOSUB 100:W1$=W$:GOSUB 100:W1$=W$+W1$
280 IF P>1 THEN B$=B$+LEFT$(TT$,P-1)
290 B$=B$+W1$+"H":IF LEN(TT$)>P THEN B$=B$+RIGHT$(TT$,LEN(TT$)-P)
295 RETURN
300 T2=DT¥&H40+16:GOSUB 50:T2=(DT MOD &H40)¥8+16:B$=B$+OP$(T2)+",": RETURN
400 T2=(DT MOD 8)+1:B$=B$+OP$(T2):RETURN
500 N=T1/100:T2=T1-N*100:T1=N:RETURN
600 SA=SB:B$=LEFT$(B$,10)+SPACE$(11)+"???":RETURN
1000 IF DT>&H7F THEN T2=DT¥8-7:GOSUB50:IF T2>12 OR T2=11 THEN 400 ELSE T2=8:GOTO
 1030
1010 IF DT=&H76 THEN T2=34:GOTO 50
1020 T2=20:GOSUB 50:T2=DT¥8-7
1030 B$=B$+OP$(T2)+",":GOTO 400
2000 GOSUB 100:IF DT<&H40 THEN T2=DT¥8+1:IF T2=7 THEN 600 ELSE GOSUB 50:GOTO 400

2010 GOSUB 300:GOTO 400
3000 Z=0:ZZ=0:QQ=0:Q=0:IF DT=&HFD THEN Y=1 ELSE Y=0
3010 GOSUB 100:IF DT=&HCB THEN 3070 ELSE IF DT<ID(20) THEN 3040
3020 FOR I=20 TO 39:IF DT=ID(I) THEN 3060 ELSE IF DT>ID(I) THEN NEXT
3030 GOTO 600
3040 FOR I=19 TO 1 STEP -1:IF DT=ID(I) THEN 3060 ELSE IF DT<ID(I) THEN NEXT
3050 GOTO 600
```

改良版PC用逆アセンブラ プログラム・リスト

```
3060 IF Y=0 THEN 3065 ELSE IF Y=1 THEN 3063
3063 Z=TD(I)/10000:Q=TD(I)/100-Z*100:IF (Z=45)OR(Z=47)OR(Z=49)THEN ZZ=10000
3064 IF (Q=45)OR(Q=47)OR(Q=49)THEN QQ=100
3065 T1=TD(I)+ZZ+QQ:GOTO 4040
3070 GOSUB 100:W1$=W$:GOSUB 100:IF DT MOD8<>6 THEN 600 ELSE IF(DT=&H6)OR(DT=&HE)
OR(DT=&H16)OR(DT=&H1E)OR(DT=&H26)OR(DT=&H2E)OR(DT=&H3E) THEN 3071 ELSE GOSUB 300
:GOTO 3080
3071 IF DT=&H6 THEN T2=1 ELSE IF DT=&HE THEN T2=2 ELSE IF DT=&H16 THEN T2=3 ELSE
 IF DT=&H1E THEN T2=4 ELSE IF DT=&H26 THEN T2=5 ELSE IF DT=&H2E THEN T2=6 ELSE I
F DT=&H3E THEN T2=8
3072 GOSUB 50
3080 B$=B$+"(I"+CHR$(&H58+Y)+"+"+W1$+"H)":RETURN
4000 B$="0000-"+SPACE$(17):W=LEN(HEX$(SA)):MID$(B$,5-W,W)=HEX$(SA):DP=6:BB$=""
4010 GOSUB 100:SB=SA:IF DT>&H3F AND DT<&HC0 THEN 1000 ELSE IF DT=&HCB THEN 2000
ELSE IF DT=&HDD OR DT=&HFD THEN 3000 ELSE IF DT=&HDC THEN T1=890221!:GOTO 4040
4020 IF DT=&HED THEN GOSUB 100:IF DT<&H40 OR DT>&HBF THEN 600
4030 T1=T(DT+1):IF T1=0 THEN 600
4040 GOSUB 500:GOSUB 50:OT2=T2
4050 IF T1=0 THEN RETURN ELSE GOSUB 500:GOSUB 200
4060 IF T1=0 THEN RETURN ELSE B$=B$+",":T2=T1:GOTO 200
5000 INPUT "START-ADRRESS";SA$
5010 INPUT "E N D -ADRRESS";EA$:IF LEN(EA$)=0 THEN 5010 ELSE IF LEN(SA$)=0 THEN
SA=0 ELSE SA=VAL("&H"+SA$):IF SA<0 THEN SA=SA+65536!
5020 EA=VAL("&H"+EA$):IF EA<0 THEN EA=EA+65536!
5030 IF SA>EA THEN 7080 ELSE GOSUB 4000:B$=B$+SPACE$(35-LEN(B$))+BB$:IF LP=0 THE
N PRINT B$ ELSE LPRINT B$
5031 K$=INKEY$:IF K$="!" THEN 7080 ELSE 5030
6000 SA=0
6010 CL=18:SA$="":PRINT:INPUT"ADRRESS-LINE";SA$:IF LEN(SA$)=0 THEN 6040 ELSE IF
SA$="!" THEN 7080 ELSE N=INSTR(SA$,"-"):IF N=0 THEN N=LEN(SA$)+1
6020 IF N>1 THEN SA=VAL("&H"+LEFT$(SA$,N-1)):IF SA<0 THEN SA=SA+65536!
6030 IF N<LEN(SA$) THEN CL=VAL(RIGHT$(SA$,LEN(SA$)-N))
6040 GOSUB 4000:B$=B$+SPACE$(35-LEN(B$))+BB$:IF LP=0 THEN PRINT B$ ELSE PRINT B$
:LPRINT B$
6041 K$=INKEY$:IF K$="!" THEN 7080 ELSE CL=CL-1:IF CL>0 THEN6040 ELSE 6010
7000 DEFINT A-Z:DEFSNG E,S,T
7010 DIM T(256),NM$(67),OP$(51),TD(39),ID(39)
7020 FOR N=  1 TO 124:READ   T(N):NEXT
7030 FOR N=161 TO 256:READ   T(N):NEXT
7040 FOR N=  1 TO  67:READ NM$(N):NEXT
7050 FOR N=  1 TO  51:READ OP$(N):NEXT
7060 FOR N=  1 TO  39:READ  TD(N):NEXT
7070 FOR N=  1 TO  39:READ  ID(N):NEXT
7080 WIDTH40:CONSOLE 0,25,0:INPUT"LPRINT";LP$:IF LP$="y" THEN LP=1 ELSE LP=0
7085 INPUT"STEP BY STEP";SA$:IF LEN(SA$)=0 THEN 6000 ELSE IF LEFT$(SA$,1)="n" TH
EN 5000 ELSE 6000
8000 DATA 000050,893320,084220,003337,000137,000129,390120,000062,383732,333509,
          420820,003329,000237,000229,390220,000065,003944,893420,084320,003437,
          000337,000329,390320,000061,003943,343509,430820,003429,000437,000429
8010 DATA 390420,000064,390943,893520,359020,003537,000537,000529,390520,000028,
          391043,353509,903520,003529,000637,000629,390620,000027,391143,893620,
          089020,003637,000737,000729,390720,000007,390243,363509,900820,003629
8020 DATA 000837,000829,390820,000022,410136,014153,333512,339020,000049,000060,
          001635,085120,410236,024153,333510,903320,000000,000059,000000,083220,
          410336,034153,343512,349020,000000,000000,001735,510820,410436,044153
8030 DATA 343510,903420,000000,000000,001835,320820,410536,054153,353512,000000,
          000000,000000,000000,000066,410636,064153,353510,000000,000000,000000,
          000000,000063,000000,000000,363512,369020,000000,000000,000000,000000
8040 DATA 410836,084153,363510,903620
8050 DATA 47,25,40,55,00,00,00,00,45,23, 38,54,00,00,00,00,48,26,41,52,
          00,00,00,00,46,24,39,51,00,00, 00,00
8060 DATA 000958,003356,890942,008942,890921,003357,390809,002467,001058,000058,
          891042,000000,891021,008921,390810,002567,001158,003456,891142,084053,
          891121,003457,003911,002667,000258,000033,890242,400836,890220,000000
8070 DATA 390812,002767,001258,003556,891242,354432,891221,003557,003913,002867,
          001358,000742,891342,353432,891321,000000,003914,002967,001458,003756,
          891442,000030,891421,003757,003915,003067,001558,353620,891542,000031
8080 DATA 891521,000000,003916,003167
8090 DATA RLC, RRC,  RL,  RR, SLA, SRA, SCF, SRL, ADD, ADC,  SUB, SBC, AND ,XOR
,  OR,  CP, BIT, RES, SET,  LD,CALL, CCF, CPD,CPDR, CPI,CPIR, CPL, DAA, DEC,  DI
,  EI,  EX, EXX,HALT
8100 DATA IM, IN, INC, IND,INDR, INI,INIR,  JP,  JR,DJNZ, LDD,LDDR, LDI,LDIR, NG
G, NOP,OTDR,OTIR, OUT,OUTD,OUTI, POP,PUSH, RET,RETI,RETN, RLA,RLCA, RLD, RRA,RRC
A, RRD, RST
8110 DATA B,   C,   D,   E,   H,   L,(HL),   A,  NZ,   Z,  NC,  PO,  PE,   P,   M
,   0,   1,   2,   3,   4,   5,   6,   7, 00H,  08H,  10H,  18H,  20H,  28H,  30
H,  38H,   R,  BC,  DE,  HL,  SP,  AF,  AF'
8120 DATA " ", ( ), (C), (BC),(DE),(SP),(IX),(IY),(IX+ ),(IY+ ), IX, IY, I
8130 DATA 334909,344909,894920,499020,004937,494909,904920,004929,004737,004729,
          394720,364909,470120,470220,470320,470420,470520,470620,014720
8140 DATA 024720,034720,044720,054720,064720,084720,470820,470809,470810,004711,
          470812,004713,004714,004715,004716,004956,494432,004957,004542,493620
8150 DATA 009,025,033,034,035,041,042,043,052,053,054,057,070,078,086,094,102,11
0,112,113,114,115,116,117,119,126,134,142,150,158,166,174,182,190,225,227,229,23
3,249
```

PC-8001用

機械語サブルーチンの作成に便利な!

《整数型》BASIC COMPILER

宮崎マイコン・ショップ　高木　秀雄

今回，開発しましたBASIC COMPILERについて全国各地から多数の問い合わせをいただきましたが，ここで誌面を借りてその概要を説明します．

1.　概　説

このコンパイラの開発の基本的思想は次のとおりです．

❶業務用ソフトウェアでBASIC言語の弱い部分，たとえばソーティングなどについてマシン語レベルの高速性を求める．

❷RAMのユーザー領域を考えて，N-BASICと完全コンパチブルなコンパイラは困難である．

もっとも，実際はINPUTやPRINTなどは，たとえば伝票をめくる時間にBASICでも対応でき，コンパイルの必要性はあまりありません．むしろ，**FOR-NEXT**, **IF-THEN**などの処理をマシン語に落とす方が，ソフトの容易性，メモリ容量などの点で有利だといえます．

2.　オブジェクトの内容

このコンパイラはBASIC記述で約20Kバイトを使用しているため，オブジェクト生成アドレスは&HD100以降約7Kバイトです（※16K用では不可）．

このオブジェクトは整数型であり，1変数あたり2バイトの独立アドレスを持っているため，BASICとのリンク変数においては整数型を使用する必要があります．また，このオブジェクトは効率の良い速度を得るため，乗算・除算以外はサブルーチンを使用しておらず，それらのサブルーチンも自動生成方式であるためZ80使用の他のマイコンにもそのまま移植することができます．

3.　使い方

ユーザーはまずソース・プログラムの準備が必要であり，これは表1の範囲における命令でなければなりません．コンパイラをRUNさせると，最初にオブジェクト・アドレス，各変数の数および変数名(配列の場合，その大きさも)を尋ねてくるのでインプットしてください．次に各コマンド（c, t, s, m, a, e）によりそれぞれの処理を行ないます．ソース・プログラムは**c**コマンドを使用し，行番号および文を1行ずつキーインしてください．打ち終わったら，コマンド**e**によりジャンプ・アドレス，サブルーチンの自動生成などのEND処理を行ない終了します．

写真1　コンパイル前に変数を定義します．

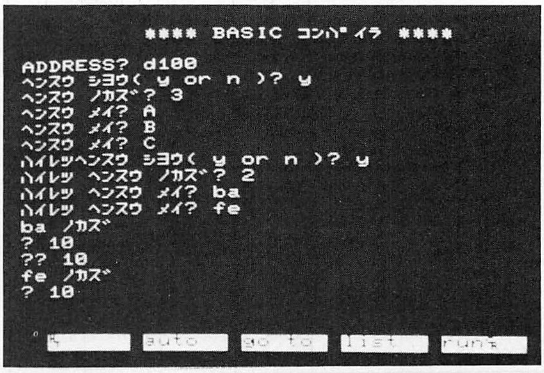
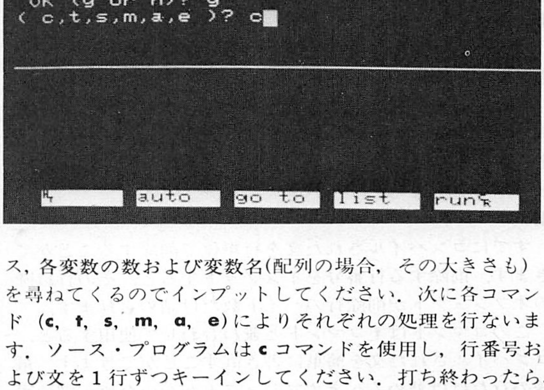


写真2　1行入力して，Y キーを押すとコンパイル



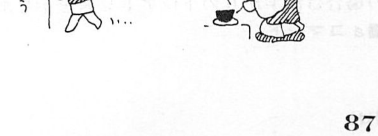


表1　使用できる命令

書　式	変 数	配 列	数 値 −32768 〜32767	16進数値 &H0000 〜&HFFFF
四則演算 ＊，／，＋，−，．	○	○	○	○
if a < b または a = b a > b または a < > b または a < = b a > = b **then**	a，b ○	a，b ○	b ○	b ○
for a = b **to** c **step** I	a, b, c ○	×	b, c，I ○	b, c，I ○
next a （a はなくてもよい）	×	×	×	×
gosub I 〜 **return**	×	×	I（行番号） ○	I（行番号） ○
a = **peek** (b)	a，b ○	a ○	b ○	b ○
poke a，b	a，b ○	×	a，b ○	a，b ○
goto I (**then**の後は省略可)	×	×	I（行番号） ○	I（行番号） ○
ldir a − b，c **lddr** a − b，c	a, b, c ○	×	×	×
cpira a − b，c = x **cpdra** a − b，c = x	a, b, c, x ○	×	x ○	x ○
ainc I **adec** I	a ○	×	I ○	×
nop I	×	×	I ○	I ○
配列の書式 **x**（a，b）または**x**（a）	a，b ○	×	a，b ○	a，b ○

○…使用可　×…使用不可

注）goto，gosub文の飛び先行番号または，FOR-NEXT文は同一コンパイル内に限られます．

4.　各コマンド説明

■cコマンド

文（ソース）1行を入力し，コンパイルします．

■tコマンド

使っていない行番号であれば文の挿入ができます．挿入バッファが100バイトしかなく，コンパイルして100バイトをオーバーするような長い文は“ソウニュウ・スペースオーバー”となる．

■sコマンド

すでにコンパイルされた文を行単位で削除することができます．削除する行番号をインプットすると，その行以降のオブジェクトが削除行のバイト数だけ詰められます．

このコマンドはtコマンドと兼ね合わせて使用すると，コンパイルされた文を簡単に書き直すことができます．最後の行だけは詰めるべきオブジェクトがないため，削除することができません．

■mコマンド

変数のアドレス・モニタで変数名をインプットすると，その変数の下位アドレスを表示します．配列変数の場合，変数名の右に”)”を付けてインプットしてください．配列の場合は添字が0の下位アドレスを示します．

■aコマンド

コンパイルした行の先頭アドレス・モニタで，行番号を入力するとそのアドレスを表示します．

■eコマンド

コンパイル終了時に入れるコマンドでジャンプ・アドレス書き込み，サブルーチンを使用する場合はそれを自動生成し，そのアドレスの書き込みを行ない，最後にオブジェクトの使用番地，バイト数およびコール・アドレスを表示します（eコマンド後は修正は不可）．

5.　注意事項

❶最初にフリー領域確保のためclearを行なってください．

```
clear 800,&HD100+α
α=(ソース内の変数の数)*3+(ソース内の配列変数の数)*7
```

これは，使用する変数の数に応じてその処理用の配列を宣言するためです（オブジェクトの先頭アドレスも&HD100＋α以降になる）．

❷各コマンド，各命令はすべて小文字で入力してください．

❸各変数名はBASICで文字列比較を行なうため，その長さは2文字以上でも良いですが，頭はA〜Zあるいはa〜zでなければいけません．また，大文字・小文字も違うものとして扱います．

❹**for-next**においては，1番新しい**for**がその次の**next**に関連しているため，挿入および削除では注意してください（**next**の後の変数は省略できます）．

❺行番号は1〜250の範囲です．また，文におけるスペースは処理時に無視されます．

❻マルチ・ステートメントは1行に20以内です．

❼四則演算以外の命令の中では式は使えません．

❽四則演算における式の評価は普通，BASICで使用するものと同じであり，そのままインプットできますが剰余だけは，ピリオド(．)を使用し，評価は加算・減算と同じにしてあります．

演算順位 （ソース・プログラム）	1	（　）（カッコ内）
	2	＊ ／（乗算・除算）
	3	＋ − ．（加算・減算・剰余）

❾その他各命令における変数・配列変数・数値・16進数値の使用可・不可が**表1**にまとめてあるので注意してください．

6.　新命令

このプログラムはオブジェクトの効率を計って次のような新命令を追加しています．

❶データの移動命令

㋑**ldir** a-b, c

㋺**ldde** a-b, c

a アドレスから，c バイトのデータをb アドレスへブロック転送します．㋑と㋺の違いはa ＞ bのときは㋑を使用し，**a ＝移動データの先頭アドレス　b ＝データが格納される先頭アドレス**となります．

もしa ＜ bのときはデータは後の方から移動させるため，㋺を使用し，

a＝移動データの後頭アドレス
b＝データが格納される後頭アドレスとなります．

❷データの比較命令

㋑cpira　a－b，c＝x
㋺cpdra　a－b，c＝x

aアドレスからbバイトのデータの中にXというデータがあればc＝1なければc＝0となります．また，c＝1の場合，aはそのデータのアドレスとなります．㋑と㋺の違いは，㋑…a＝ブロック·データの先頭アドレス　㋺…a＝ブロック·データの後頭アドレスを示すことです．

❸変数インクリメント，デクリメント

㋑ainc I（Iは0～9までの数）
㋺adec I（　〃　）

変数aをIだけ加えたり減じたりするものでa＝a±Iと等価ですが，Iが3以下だとこの方が効率が良いです．

❹nop I

命令ではなく，コンパイル中にアドレスをI（数値のみ）進めるものです．

注）新命令❶❷についてはa,b,cをあらかじめ定義しておいていてください．Xについては，変数（配列は不可），数値が使用用できます．❸についてはaが変数，Iは数値のみです．

7. エラーメッセージ

❶ヘンスウナシ!!

最初に宣言した変数名以外を使用のとき，文法を間違ったとき．

❷ギョウニュウリョクMISS!!

c，tコマンドにおいて，すでに使用された行番号をインプットしたとき．sコマンドで1番高い行番号をインプットしたとき．

❸メモリ・オーバー

オブジェクトがフリー領域よりオーバーしたとき．

❹ソウニュウ・スペースオーバー

tコマンドでコンパイルされた1行あたりのオブジェクトが100バイトをオーバーしたとき．

❺if-MISS

Ifの比較文に，=，<，>を使っていないとき．

❻poke-MISS

poke文に，コンマ（，）がないとき．

❼next-MISS

forが使用される前にnextを使用したとき．

❽？？MISS

行の文がヌル・ストリングのとき．

❾if-jmp MISS!!

最後のEND処理でサーチするものでIf文の次の行がないとき（If文で最後の行が終了したとき）．

❿NOT no I（Iは行番号）

最後のEND処理でサーチするもので，ジャンプ先の行が使用されていないとき．この場合，END処理は最後まで実行（サブルーチンの行がないときも同様）．

⓫アドレスナシ!!

aコマンドで使用されていない行番号をインプットしたとき．

⓬BA?? MISS

パラメータ誤りなどでBASICインタープリタ内でエラーが発生したとき（行番号が250をオーバーしたときなど…）

注）❾❿を除くそれぞれのエラーはその文をインプットする前のコマンド・インプット状態へ戻ります．

8. 実例

リスト1は変数A，B，C，Dがそれぞれ1からYまで変化するとき，組み合わせの数を求めるもので，これをこのままRUNさせるとY＝13のとき，19分51秒かかります．

次に，このプログラムをコンパイルしてみます．コンパイルするのは20行から140行までで，この部分が機械語サブルーチンとして置き換わるわけです．

70，80，90，100行のAND文はコンパイラの書式にないので書き直して整理すると，ソース・プログラムはリスト2のようになります．

行番号は250以内であれば，そのまま使用できますが整理するため付け直しました．1行目のt＝0はオブジェクト内においては変数の初期状態が不定であるためです．15行目のreturnは機械語サブルーチンをBASICへ戻すためのもので，ソース内で使用するサブルーチンのreturnと同じものです．なお，ソース内の各変数は大文字・小文字それぞれ違うものとして扱います．

ここではすべての変数に大文字を使用しました．なお，BASICとのリンク変数はTとYです．まず使用する変数がA，B，C，D，T，Y，C2，C2，C3の9個であるためclear文を次のように実行します．

```
clear 800, &H(D100+3*9以上)
```

次にコンパイラをRUNさせると，

```
ADDRESS?
```

と表示します．

リスト1　初期プログラム

```
5 DEFINT A-Z
10 INPUT Y
20 FOR A=1 TO Y
30 FOR B=1 TO Y
40 FOR C=1 TO Y
50 FOR D=1 TO Y
60 C1=0:C2=0:C3=0
70 IF A<>B AND A<>C THEN C1=1
80 IF A<>D AND B<>C THEN C2=1
90 IF B<>D AND C<>D THEN C3=1
100 IF C1=1 AND C2=1 AND C3=1 THEN T=T+1
110 NEXT
120 NEXT
130 NEXT
140 NEXT
150 PRINT T
160 END
```

リスト2　ソース・プログラム

```
1 T=0
2 FOR A=1 TO Y
3 FOR B=1 TO Y
4 FOR C=1 TO Y
5 FOR D=1 TO Y
6 C1=0:C2=0:C3=0
7 IF A<>B THEN IF A<>C THEN C1=1
8 IF A<>D THEN IF B<>C THEN C2=1
9 IF B<>D THEN IF C<>D THEN C3=1
10 IF C1=1 THEN IF C2=1 THEN IF C3=1 THEN T=T+1
11 NEXT
12 NEXT
13 NEXT
14 NEXT
15 RETURN
```

リスト3　コンパイルされたオブジェクトをコールするプログラム

```
5 DEFINT A-Z
10 INPUT Y
20 POKE &HE00A,PEEK(VARPTR(Y)):POKE &HE00B,PEEK(VARPTR(Y)+1)
30 DEFUSR0=&HE026:AA=USR0(0)
40 T=0
50 POKE VARPTR(T),PEEK(&HE008):POKE VARPTR(T)+1,PEEK(&HE009)
60 PRINT T
70 END
```

これは変数エリアの先頭アドレスのことで，フリー領域以降をインプットしてください（フリー領域以外をインプットしてもよい）．ここではE000をインプットします．

ヘンスウシヨウ（y or n）？

と表示します．

これは変数を使用するかどうかを尋ねているので Y をインプットします．次に，

ヘンスウノカズ？

と表示します．

9とインプットしてください．すると次に，

ヘンスウメイ？

と表示します．

これは9回(変数の数だけ)繰り返えすので，変数名をインプットします（インプットした順に，変数アドレスが形成される）．

ここではA，B，C，D，T，Y，C1，C2，C3，の順にインプットします．次に，

ハイレツヘンスウシヨウ（y or n）？

と表示します．ここでは配列を使用していないので，nをインプットするのですが，もし使用する場合は次のようになります．

1．ハイレツヘンスウノカズ？
2．ハイレツヘンスウメイ？
3．(ハイレツヘンスウメイ) ノカズ？

1．2．は上記の変数の場合と同じですが，3．については添字数の宣言を尋ねているので，そのインプット方式は，

1次元の場合は　I，0
2次元の場合は　I1，I2　（＊I，I1，I2は配列の箱の数）

とします（なお，使用可は2次元まで）．これを配列変数名の数だけ繰り返します．

注) BASICの場合，DIMX(5)と宣言すると，箱の数は5＋1の6個となりますが，ここではI，またはI1×I2がそのまま箱の数となります．

変数アドレスが決定されると，さらにそれに20バイトの作業領域を加えたアドレスを表示し，ここをサブルーチンのコール・アドレス（clear 領域内でなければならない）とします．

E026
SUB－アドレスOK（y or n）
(nをインプットすると,コール･アドレスを変えることができる.)

次はコマンド・モードに移り，**c**コマンドによりソース・プログラムをインプットしてください．コマンドは1行ごとにインプット後，確認のためもう一度その文を表示し尋ねてくるので，OKであれば Y をインプットします．以上15行までインプットします．

次にBASICとのリンク変数であるTとYのアドレスを**m**コマンドで調べます．

ここではT＝E008，Y＝E00Aとなります．最後に**e**コマンドでEND処理を行ないます．以下を表示してENDとなります．

シヨウメモリE000－E130　305バイト
サブルーチンアドレスE026

このオブジェクトをBASICの機械語サブルーチンとしてコールするわけです．リンクすべき変数はBASICのT，YとオブジェクトのT，Yであるから，これにはN-BASICのVARPTRを使い，プログラムを整理すると**リスト3**のようになります．

このプログラムでRUNさせると，Y＝13でわずか8秒であり，約150倍ほど速度を高めることができたことになります．

注) プリンタを使用しないときは，次のリストの ▭ の部分を削除してください．

3870 [LPRINT Y$; HI:] M1＝HA(M2)－HA(HI)
:M4＝DC－HA(M2)：R9＝HA(M2)
5010 [LPRINT Y$; HI;"] "; Q$

9.　最後に

いままでの説明で充分におわかりいただけたものと思います．ぜひこのコンパイラを多くの人が利用しソフトの効率化を押し進めてくれるよう希望します．

BASICコンパイラ　プログラム・リスト

```
10 REM < clear 800,d000+@ >
20 ON ERROR GOTO 0
30 PRINT CHR$(12)
40 DEFINT A-Z:WIDTH 40,20
50 CONSOLE 2,20,1,0:GOSUB 5050
60 DEFFNA(K,J)=VAL("&H"+MID$(HEX$(K),J,2))
70 KZ=&HE990:KM=KZ:HQ=1:HM=1:LB=1:LC=1:LD=1
80 DIM H1$(20),HA(250),HB(100),HC(100),HE$(10),HD(10),HF$(10),GA$(10),HG$(20),HJ
$(20),Z$(20)
90 PRINT :INPUT"ADDRESS";DD$:DC=VAL("&H"+DD$)
100 INPUT"ﾍﾝｽｳ ｼﾖｳ( y or n )";Y$
110 IF Y$="n" THEN 140
120 INPUT"ﾍﾝｽｳ ﾉｶｽﾞ";VA:DIM EA$(VA):EF=VA:EB=DC
130 FOR D1=1 TO VA:INPUT"ﾍﾝｽｳ ﾒｲ";VN$:EA$(D1)=VN$+"":DC=DC+2:NEXT
140 INPUT"ﾊｲﾚﾂﾍﾝｽｳ ｼﾖｳ( y or n )";Y$
150 IF Y$="n" THEN 200
160 INPUT"ﾊｲﾚﾂ ﾍﾝｽｳ ﾉｶｽﾞ";VA:DIM FA$(VA),FB(VA),FC(VA):FE=VA
170 FOR D1=1 TO VA:INPUT"ﾊｲﾚﾂ ﾍﾝｽｳ ﾒｲ";VN$:FA$(D1)=VN$+"":NEXT
180 FOR D1=1 TO VA:PRINT FA$(D1);" ﾉｶｽﾞ":INPUT VB,VC
190 IF VC=0 THEN FB(D1)=DC:DC=DC+2*VB:NEXT D1 ELSE FB(D1)=DC:DC=DC+2*VC*VB:FC(D1
)=VC:VC=0:NEXT D1
200 ZX$=HEX$(DC):DC=DC+20:ST$=HEX$(DC)
210 PRINT ST$
220 INPUT "SUB-ｱﾄﾞﾚｽ OK ( y or n )";Y$
230 IF Y$="n" THEN INPUT "ｱﾄﾞﾚｽ";ST$:DC=VAL("&H"+ST$)
240 REM < START >
250 ON ERROR GOTO 3410
260 PRINT CHR$(12):GOSUB 5050
270 PRINT :PRINT
280 IF DC>KM THEN PRINT "ﾒﾓﾘ ｵｰﾊﾞｰ":GOTO 3420
290 HZ=HI
300 MX=DC:MB=LB:MC=LC:MD=LD
310 IP=0:INPUT "( c,t,s,m,a,e )";Y$
320 ON INSTR("ctsmae",Y$)GOTO 330,3460,3810,3240,3300,3340
330 ERASE H1$:DIM H1$(20):HQ=1:HM=1
340 INPUT"ｷﾞｮｳ ﾊﾞﾝｺﾞｳ";HI
350 IF HA(HI)<>0 OR(IP=0 AND HI<HZ) THEN 3450
360 IF IP=1 THEN 3480
370 HA(HI)=DC
380 GOSUB 4930:GOTO 620
390 FOR D1=1 TO HQ-1
400 IF H1$(D1)="" THEN PRINT "?? MISS":GOTO 3420
410 IF INSTR(H1$(D1),"if")<>0 THEN 690
420 IF INSTR(H1$(D1),"for")<>0 THEN 1100
430 IF INSTR(H1$(D1),"goto")<>0 THEN 1240
440 IF INSTR(H1$(D1),"peek")<>0 THEN 1280
450 IF INSTR(H1$(D1),"poke")<>0 THEN 1400
460 IF INSTR(H1$(D1),"next")<>0 THEN 1510
470 IF INSTR(H1$(D1),"gosub")<>0 THEN 2360
480 IF INSTR(H1$(D1),"return")<>0 THEN 2410
490 IF INSTR(H1$(D1),"inc")<>0 THEN R9=&H23:GOTO 2440
500 IF INSTR(H1$(D1),"dec")<>0 THEN R9=&H2B:GOTO 2440
510 IF INSTR(H1$(D1),"nop")<>0 THEN 2510
520 IF LEFT$(H1$(D1),4)="ldir" THEN R9=&HB0:GOTO 2560
530 IF LEFT$(H1$(D1),4)="lddr" THEN R9=&HB8:GOTO 2560
540 IF INSTR(H1$(D1),"cpir")<>0 THEN R9=&HB1:GOTO 2650
550 IF INSTR(H1$(D1),"cpdr")<>0 THEN R9=&HB9:GOTO 2650
560 IF H1$(D1)<":" THEN I1$=H1$(D1)+"":GOTO 1260
570 GOTO 1660
580 IF EE=1 THEN EE=0:GOTO 3420
590 NEXT D1
600 IF IP=1 THEN 3530
610 GOTO 280
620 REM < h$ ﾉ ﾌﾞﾝｶｲ >
630 R9=INSTR(HM,H$,":")
640 IF R9=0 THEN H1$(HQ)=RIGHT$(H$,LEN(H$)-HM+1):GOTO 660
650 H1$(HQ)=MID$(H$,HM,R9-HM)
660 HQ=HQ+1:HM=R9+1
670 IF R9=0 THEN H$="":GOTO 390
680 GOTO 630
690 REM < if ﾉ ﾍﾝｶﾝ >
700 H4=INSTR(H1$(D1),"then")
710 I1$=MID$(H1$(D1),3,H4-3)
720 FOR E2=1 TO LEN(I1$):I2$=MID$(I1$,E2,1)
730 IF I2$="=" OR I2$="<" OR I2$=">" THEN I2$=LEFT$(I1$,E2-1):GOTO 750 ELSE NEXT
 E2
740 PRINT "if-MISS":GOTO 3420
750 FOR E3=LEN(I1$) TO 1 STEP-1:I3$=MID$(I1$,E3,1)
760 IF I3$="=" OR I3$="<" OR I3$=">" THEN I3$=RIGHT$(I1$,LEN(I1$)-E3):GOTO 770 E
LSE NEXT E3
770 IF LEN(I1$)-LEN(I2$)-LEN(I3$)=2 THEN IL=1:GOTO 790
780 IL=INSTR("=<>",MID$(I1$,E2,1))+1
790 IF RIGHT$(I2$,1)<>")" THEN 840
800 R9=INSTR(I2$,"("):HJ$(0)=LEFT$(I2$,R9-1)
810 HG$(0)=MID$(I2$,R9+1,LEN(I2$)-R9-1)
820 TG=0:II=0:GOSUB 2190
830 IF SM=0 THEN POKE DC,&HEB:DC=DC+1:GOTO 860 ELSE 860
840 G3$=I2$+"":GOSUB 2830:R1=HH
850 POKE DC,&H2A:POKE DC+1,FNA(R1,3):POKE DC+2,FNA(R1,1):DC=DC+3
860 IF RIGHT$(I3$,1)<>")" THEN 910
870 R9=INSTR(I3$,"("):HJ$(0)=LEFT$(I3$,R9-1)
```

BASICコンパイラ　プログラム・リスト

```
880 HG$(0)=MID$(I3$,R9+1,LEN(I3$)-R9-1)
890 TG=0:II=1:GOSUB 2190
900 IF SM=0 THEN POKE DC,&HE1:DC=DC+1:GOTO 940 ELSE 940
910 IF I3$<":" THEN R2=VAL(I3$):POKE DC,&H11 ELSE G3$=I3$+"":GOSUB 2830:R2=HH:PO
KE DC,&HED:POKE DC+1,&H5B:DC=DC+1
920 IF R2>=0 THEN POKE DC+1,R2MOD256:POKE DC+2,R2¥256:DC=DC+3:GOTO 940
930 POKE DC+1,FNA(R2,3):POKE DC+2,FNA(R2,1):DC=DC+3
940 I2$=MID$(I1$,E2,E3-E2+1)
950 IF IL=1 AND I2$="<>" THEN IL=1
960 IF IL=1 AND(I2$="<=" OR I2$="=<") THEN IL=5
970 IF IL=1 AND(I2$=">=" OR I2$="=>") THEN IL=6
980 IF IL<1 OR IL>6 THEN 740
990 IF IL=1 THEN GOSUB 1090:POKE DC+3,&HCA
1000 IF IL=2 THEN GOSUB 1090:POKE DC+3,&HC2
1010 IF IL=3 THEN GOSUB 1090:POKE DC+3,&HF2
1020 IF IL=4 THEN POKE DC,&HEB:DC=DC+1:GOSUB 1090:POKE DC+3,&HF2
1030 IF IL=5 THEN POKE DC,&HEB:DC=DC+1:GOSUB 1090:POKE DC+3,&HFA
1040 IF IL=6 THEN GOSUB 1090:POKE DC+3,&HFA
1050 POKE DC+4,HI:POKE DC+5,&HFF
1060 HB(LB)=DC+4:LB=LB+1:DC=DC+6
1070 H1$(D1)=RIGHT$(H1$(D1),LEN(H1$(D1))-H4-3)
1080 GOTO 400
1090 POKE DC,&HAF:POKE DC+1,&HED:POKE DC+2,&H52:RETURN
1100 REM < for ﾉ ﾍﾝｶﾝ >
1110 R9=INSTR(H1$(D1),"to"):H4=INSTR(H1$(D1),"step")
1120 IF H4<>0 THEN GA$(LD)=RIGHT$(H1$(D1),LEN(H1$(D1))-H4-3)
1130 I1$=MID$(H1$(D1),4,R9-4)
1140 IF H4=0 THEN I2$=RIGHT$(H1$(D1),LEN(H1$(D1))-R9-1) ELSE I2$=MID$(H1$(D1),R9
+2,H4-R9-2)
1150 HE$(LD)=LEFT$(I1$,INSTR(I1$,"=")-1):HD(LD)=DC+6
1160 HF$(LD)=I2$+""
1170 G3$=HE$(LD)+"":GOSUB 2830:R1=HH:LD=LD+1
1180 I3$=RIGHT$(I1$,LEN(I1$)-INSTR(I1$,"="))
1190 IF I3$<":" THEN R2=VAL(I3$):POKE DC,&H21 ELSE G3$=I3$+"":GOSUB 2830:R2=HH:P
OKE DC,&H2A
1200 IF R2>=0 THEN POKE DC+1,R2MOD256:POKE DC+2,R2¥256:GOTO 1220
1210 POKE DC+1,FNA(R2,3):POKE DC+2,FNA(R2,1)
1220 POKE DC+3,&H22:POKE DC+4,FNA(R1,3):POKE DC+5,FNA(R1,1)
1230 DC=DC+6:GOTO 580
1240 REM < goto ﾉ ﾍﾝｶﾝ >
1250 I1$=RIGHT$(H1$(D1),LEN(H1$(D1))-4)
1260 R1=VAL(I1$):POKE DC,&HC3:POKE DC+1,R1MOD256:POKE DC+2,R1¥256:HB(LB)=DC+1:LB
=LB+1
1270 DC=DC+3:GOTO 580
1280 REM < peek ﾉ ﾍﾝｶﾝ >
1290 R9=INSTR(H1$(D1),"peek"):I2$=MID$(H1$(D1),R9+5,LEN(H1$(D1))-R9-5)
1300 I3$=LEFT$(H1$(D1),R9-2)
1310 IF I2$<":" THEN R1=VAL(I2$) ELSE G3$=I2$+"" :GOSUB 2830:R1=HH:CC=1
1320 POKE DC,&H3A:IF R1<0 THEN POKE DC+1,FNA(R1,3):POKE DC+2,FNA(R1,1) ELSE POKE
 DC+1,R1MOD256:POKE DC+2,R1¥256
1330 IF CC=1 THEN CC=0:POKE DC,&H2A:POKE DC+3,&H7E:DC=DC+1
1340 IF RIGHT$(I3$,1)<>")" THEN 1380
1350 R9=INSTR(I3$,"("):HJ$(0)=LEFT$(I3$,R9-1):HG$(0)=MID$(I3$,R9+1,LEN(I3$)-R9-1
)
1360 TG=0:DC=DC+3:II=0:GOSUB 4090:GOSUB 2220
1370 POKE DC,&H77:POKE DC+1,&H23:POKE DC+2,&HAF:POKE DC+3,&H77:DC=DC+4:GOTO 580
1380 G3$=I3$+"":GOSUB 2830:R2=HH
1390 POKE DC+3,&H21:POKE DC+4,FNA(R2,3):POKE DC+5,FNA(R2,1):DC=DC+6:GOTO 1370
1400 REM < poke ﾉ ﾍﾝｶﾝ >
1410 R9=INSTR(H1$(D1),",")
1420 IF R9=0 THEN PRINT "poke-MISS":GOTO 3420
1430 I1$=MID$(H1$(D1),5,R9-5)
1440 I2$=RIGHT$(H1$(D1),LEN(H1$(D1))-R9)
1450 IF I2$<":" THEN R2=VAL(I2$):II=1 ELSE G3$=I2$+"":GOSUB 2830:R2=HH
1460 IF II=1 THEN POKE DC,&H3E:POKE DC+1,R2:POKE DC+2,0:II=0 ELSE POKE DC,&H3A:P
OKE DC+1,FNA(R2,3):POKE DC+2,FNA(R2,1)
1470 IF I1$<":" THEN R1=VAL(I1$) ELSE G3$=I1$+"":GOSUB 2830:R1=HH:CC=1
1480 POKE DC+3,&H32:POKE DC+4,FNA(R1,3):POKE DC+5,FNA(R1,1)
1490 IF CC=1 THEN CC=0:POKE DC+3,&H2A:POKE DC+6,&H77:DC=DC+1
1500 DC=DC+6:GOTO 580
1510 REM < next ﾉ ﾍﾝｶﾝ >
1520 IF LD<2 THEN PRINT "next-MISS":GOTO 3420
1530 D2=LD-1:LD=LD-1
1540 I1$=HE$(D2)+"":I2$=HF$(D2)+"":R3=HD(D2):R9=VAL(GA$(D2)):GOTO 1550
1550 G3$=I1$+"":GOSUB 2830:R1=HH
1560 IF I2$<":" THEN R2=VAL(I2$):POKE DC,&H21 ELSE G3$=I2$+"":GOSUB 2830:R2=HH:P
OKE DC,&H2A
1570 IF R2>=0 THEN POKE DC+1,R2MOD256:POKE DC+2,R2¥256:POKE DC+3,&HEB:GOTO 1590
1580 POKE DC+1,FNA(R2,3):POKE DC+2,FNA(R2,1):POKE DC+3,&HEB
1590 POKE DC+4,&H2A:POKE DC+5,FNA(R1,3):POKE DC+6,FNA(R1,1)
1600 POKE DC+7,&HE5:POKE DC+8,&HAF:POKE DC+9,&HED:POKE DC+10,&H52:POKE DC+11,&HE
1:DC=DC+5
1610 IF R9<0 THEN POKE DC+7,&H1:POKE DC+8,FNA(R9,3):POKE DC+9,FNA(R9,1):POKE DC+
10,&H9:DC=DC+3:GOTO 1630
1620 IF R9=0 THEN POKE DC+7,&H23 ELSE POKE DC+7,&H1:POKE DC+8,R9MOD256:POKE DC+9
,R9¥256:POKE DC+10,&H9:DC=DC+3
1630 POKE DC+8,&H22:POKE DC+9,FNA(R1,3):POKE DC+10,FNA(R1,1)
1640 POKE DC+11,&HC2:POKE DC+12,FNA(R3,3):POKE DC+13,FNA(R3,1)
1650 DC=DC+14:GOTO 580
1660 REM < ｼﾂｸ ﾉ ﾍﾝｶﾝ >
1670 TG=1
```

```
1680 FOR D2=1 TO LEN(H1$(D1)):I1$=MID$(H1$(D1),D2,1):I2$=MID$(H1$(D1),D2+1,1)
1690 IF I1$="" THEN GOTO 1730
1700 IF I1$="(" AND I2$="-" THEN H1$(D1)=LEFT$(H1$(D1),D2)+"0"+RIGHT$(H1$(D1),LE
N(H1$(D1))-D2):GOTO 1720
1710 IF((I1$>"@")OR (I1$>"/" AND I1$<":"))AND(I2$="(") THEN GOSUB 1820
1720 NEXT D2
1730 R9=INSTR(H1$(D1),"="):I1$=LEFT$(H1$(D1),R9-1)
1740 Z9$=RIGHT$(H1$(D1),LEN(H1$(D1))-R9):GOSUB 4290:GOSUB 1900
1750 IF LEFT$(I1$,1)<>"○" THEN 1790
1760 TG=VAL(RIGHT$(I1$,2))-10:II=1:GOSUB 4090
1770 IF (SS=1 AND SL=1)OR(SI=0 AND SS=1) THEN POKE DC-4,&H22:POKE DC-3,PEEK(DC-2
):POKE DC-2,PEEK(DC-1):DC=DC-1:SM=0:II=0:SS=0:SL=0:SI=0:GOTO 580
1780 POKE DC,&HD1:POKE DC+1,&H73:POKE DC+2,&H23:POKE DC+3,&H72:DC=DC+4:GOTO 1815

1790 G3$=I1$+"":GOSUB 2830:R1=HH
1800 POKE DC,&H22:POKE DC+1,FNA(R1,3):POKE DC+2,FNA(R1,1)
1810 DC=DC+3:GOTO 1815
1815 SM=0:II=0:SS=0:SL=0:SI=0:GOTO 580
1820 FOR E2=D2 TO 1 STEP-1:I3$=MID$(H1$(D1),E2,1)
1830 IF (I3$>"'" AND I3$<"0")OR I3$="=" THEN 1840 ELSE NEXT E2
1840 FOR E3=D2 TO LEN(H1$(D1)):I3$=MID$(H1$(D1),E3,1)
1850 IF I3$=")" THEN E3=E3+1:GOTO 1860 ELSE NEXT E3
1860 I3$=MID$(H1$(D1),E2+1,E3-E2-1):R9=INSTR(I3$,"(")
1870 HJ$(TG)=LEFT$(I3$,R9-1):HG$(TG)=MID$(I3$,R9+1,LEN(I3$)-R9-1)
1880 W2$=LEFT$(H1$(D1),E2):W3$=RIGHT$(H1$(D1),LEN(H1$(D1))-E3+1)
1890 H1$(D1)=W2$+"○"+RIGHT$(STR$(10+TG),2)+W3$:TG=TG+1:RETURN
1900 REM < ｼｯｸ SUB-1 >
1910 IG=0
1920 FOR D2=0 TO TX
1930 IF Z$(D2)="" THEN 2050
1940 R9=INSTR(Z$(D2),"・")
1950 IF R9=0 THEN IG=1:I2$=Z$(D2)+"" ELSE I2$=LEFT$(Z$(D2),R9-1)
1960 IF LEFT$(I2$,1)="○" THEN II=0:GOSUB 2170:IF SM=1 THEN SM=0:GOTO 2060 ELSE P
OKE DC,&HEB:DC=DC+1:GOTO 2060
1970 IF LEFT$(I2$,1)="●" THEN GOSUB 2310:GOTO 1990
1980 GOSUB 2230
1990 IF IG=1 THEN 2030
2000 H4=R9:FOR E2=R9+1 TO LEN(Z$(D2)):I4$=MID$(Z$(D2),E2,1)
2010 IF I4$>")" AND I4$<"0" THEN GOSUB 2070:H4=E2
2020 NEXT E2
2030 R3=VAL("&H"+ZX$)+2*D2
2040 POKE DC,&H22:POKE DC+1,FNA(R3,3):POKE DC+2,FNA(R3,1):DC=DC+3
2050 NEXT D2:DC=DC-3:RETURN
2060 IF IG=1 THEN IG=0:GOTO 2030 ELSE 2000
2070 REM < ｼｯｸ SUB-2 >
2080 I2$=MID$(Z$(D2),H4+1,E2-H4-1)
2090 IF LEFT$(I2$,1)="○" THEN II=1:GOSUB 2170:IF SM=2 THEN SM=0:GOTO 2120 ELSE P
OKE DC,&HE1:DC=DC+1:GOTO 2120
2100 IF LEFT$(I2$,1)="●" THEN SS=1:GOSUB 2310:GOTO 2120
2110 SS=1:GOSUB 2230
2120 IL=INSTR("*/+-.",MID$(Z$(D2),E2,1))
2130 IF IL=3 THEN POKE DC,&H19:DC=DC+1:RETURN
2140 IF IL=4 THEN POKE DC,&HAF:POKE DC+1,&HED:POKE DC+2,&H52:DC=DC+3:RETURN
2150 POKE DC,&HCD:POKE DC+1,IL:POKE DC+2,0:HC(LC)=DC+1:LC=LC+1
2160 IF IL=5 THEN POKE DC+3,&HC5:POKE DC+4,&HE1:DC=DC+5:RETURN ELSE DC=DC+3:RETU
RN
2170 REM < ｼｯｸ SUB-3 >
2180 TG=VAL(RIGHT$(I2$,2))-10
2190 SM=0:GOSUB 4090
2200 IF (SS=1 AND SL=1)OR(SI=0 AND SS=1) THEN IF II=0 THEN POKE DC-3,&H2A:SM=1:G
OTO 2220 ELSE POKE DC-4,&HED:POKE DC-3,&H5B:SM=2:GOTO 2220
2210 POKE DC,&H5E:POKE DC+1,&H23:POKE DC+2,&H56:DC=DC+3
2220 II=0:SI=0:SS=0:SL=0:RETURN
2230 REM < ｼｯｸ SUB-4 >
2240 IF I2$<":" THEN II=1:R1=VAL(I2$) ELSE G3$=I2$+"":GOSUB 2830:R1=HH
2250 IF II=1 AND SS=1 THEN POKE DC,&H11:IF R1<0 THEN 2290 ELSE 2300
2260 IF II=1 AND SS=0 THEN POKE DC,&H21:IF R1<0 THEN 2290 ELSE 2300
2270 IF II=0 AND SS=1 THEN POKE DC,&HED:POKE DC+1,&H5B:DC=DC+1
2280 IF II=0 AND SS=0 THEN POKE DC,&H2A
2290 POKE DC+1,FNA(R1,3):POKE DC+2,FNA(R1,1):DC=DC+3:SS=0:II=0:RETURN
2300 POKE DC+1,R1MOD256:POKE DC+2,R1¥256:DC=DC+3:SS=0:II=0:RETURN
2310 REM < ｼｯｸ SUB-5 >
2320 TG=VAL(RIGHT$(I2$,2))-10:R1=VAL("&H"+ZX$)+2*TG
2330 IF SS=0 AND PEEK(DC-1)=FNA(R1,1) AND PEEK(DC-2)=FNA(R1,3) THEN DC=DC-3:RETU
RN
2340 IF SS=1 THEN SS=0:POKE DC,&HED:POKE DC+1,&H5B:DC=DC+1 ELSE POKE DC,&H2A
2350 POKE DC+1,FNA(R1,3):POKE DC+2,FNA(R1,1):DC=DC+3:RETURN
2360 REM < gosub ﾉ ﾍﾝｶﾝ >
2370 I1$=RIGHT$(H1$(D1),LEN(H1$(D1))-5)
2380 R1=VAL(I1$)
2390 POKE DC,&HCD:POKE DC+1,R1MOD256:POKE DC+2,R1¥256
2400 HB(LB)=DC+1:LB=LB+1:DC=DC+3:GOTO 580
2410 REM < return ﾉ ﾍﾝｶﾝ >
2420 POKE DC,&HC9:DC=DC+1:GOTO 580
2430 REM < inc,dec SUB >
2440 H4=VAL(RIGHT$(H1$(D1),1)):I1$=LEFT$(H1$(D1), LEN(H1$(D1))-4)
2450 G3$=I1$+"":GOSUB 2830:R2=HH
2460 POKE DC,&H2A:POKE DC+1,FNA(R2,3):POKE DC+2,FNA(R2,1)
2470 DC=DC+3
2480 FOR E2=1 TO H4:POKE DC,R9:DC=DC+1:NEXT E2
2490 POKE DC,&H22:POKE DC+1,FNA(R2,3):POKE DC+2,FNA(R2,1)
```

BASICコンパイラ　プログラム・リスト

```
2500 DC=DC+3:GOTO 580
2510 REM < nop ﾉ ﾍﾝｶﾝ >
2520 I1$=RIGHT$(H1$(D1),LEN(H1$(D1))-3)
2530 IF I1$<":" THEN R1=VAL(I1$) ELSE PRINT "nop-MISS":GOTO 3420
2540 FOR E2=0 TO R1-1:POKE DC+E2,0:NEXT E2
2550 DC=DC+R1:GOTO 580
2560 REM < ldir,lddr >
2570 IF R9=&HB0 THEN E1=INSTR(H1$(D1),">") ELSE E1=INSTR(H1$(D1),"<")
2580 GOSUB 2800
2590 G3$=I3$+"":GOSUB 2830:R3=HH
2600 POKE DC,&H2A:POKE DC+1,FNA(R3,3):POKE DC+2,FNA(R3,1)
2610 POKE DC+3,&HE5:POKE DC+4,&HC1
2620 POKE DC+5,&H2A:POKE DC+6,FNA(R2,3):POKE DC+7,FNA(R2,1):POKE DC+8,&HEB
2630 POKE DC+9,&H2A:POKE DC+10,FNA(R1,3):POKE DC+11,FNA(R1,1)
2640 POKE DC+12,&HED:POKE DC+13,R9:DC=DC+14:GOTO 580
2650 REM < cpir,cpdr >
2660 E1=INSTR(H1$(D1),"-")
2670 GOSUB 2800
2680 H4=INSTR(I3$,"="):I1$=RIGHT$(I3$,LEN(I3$)-H4):I2$=LEFT$(I3$,H4-1)
2690 G3$=I2$+"":GOSUB 2830:R3=HH
2700 IF I1$<":" THEN H4=VAL(I1$):GOTO 2710 ELSE G3$=I1$+"":GOSUB 2830:H4=HH:GOTO
 2720
2710 POKE DC,&H3E:POKE DC+1,H4:DC=DC+2:GOTO 2730
2720 POKE DC,&H3A:POKE DC+1,FNA(H4,3):POKE DC+2,FNA(H4,1):DC=DC+3
2730 POKE DC,&H2A:POKE DC+1,FNA(R2,3):POKE DC+2,FNA(R2,1):POKE DC+3,&HE5:POKE DC
+4,&HC1
2740 POKE DC+5,&H2A:POKE DC+6,FNA(R1,3):POKE DC+7,FNA(R1,1):POKE DC+8,&HED:POKE
DC+9,R9
2750 IF R9=&HB1 THEN POKE DC+10,&H2B ELSE POKE DC+10,&H23
2760 POKE DC+11,&H22:POKE DC+12,FNA(R1,3):POKE DC+13,FNA(R1,1):DC=DC+4
2770 POKE DC+10,&H21:POKE DC+11,&HFF:POKE DC+12,&HFF:POKE DC+13,&H20:POKE DC+14,
&H1:POKE DC+15,&H23:POKE DC+16,&H23
2780 POKE DC+17,&H22:POKE DC+18,FNA(R3,3):POKE DC+19,FNA(R3,1)
2790 DC=DC+20:GOTO 580
2800 E2=INSTR(H1$(D1),","):E3=LEN(H1$(D1))
2810 I1$=MID$(H1$(D1),5,E1-5):I2$=MID$(H1$(D1),E1+1,E2-E1-1):I3$=RIGHT$(H1$(D1),
E3-E2)
2820 G3$=I1$+"":GOSUB 2830:R1=HH:G3$=I2$+"":GOSUB 2830:R2=HH:RETURN
2830 REM < ﾍﾝｽｳ ｱﾄﾞﾚｽ >
2840 FOR E1=1 TO EF
2850 IF EA$(E1)=G3$ THEN HH=EB+2*(E1-1):RETURN
2860 NEXT E1:PRINT "ﾍﾝｽｳ ﾅｼ!!":EE=1:RETURN
2870 REM < ﾊｲﾚﾂﾍﾝｽｳ ｱﾄﾞﾚｽ >
2880 FOR E1=1 TO FE
2890 IF FA$(E1)=G6$ THEN HH=FB(E1):RETURN
2900 NEXT E1:PRINT "ﾍﾝｽｳ ﾅｼ!!":EE=1:E1=0:RETURN
2910 REM < ｼﾖｳ SUB ｶｷｺﾐ >
2920 IF LC=1 THEN RETURN
2930 FOR E1=1 TO LC-1
2940 IF HC(E1)=0 THEN 2970
2950 IF PEEK(HC(E1))=1 THEN 2980
2960 IF PEEK(HC(E1))=2 OR PEEK(HC(E1))=5 THEN 3030
2970 NEXT E1:RETURN
2980 POKE HC(E1),FNA(DC,3):POKE HC(E1)+1,FNA(DC,1)
2990 IF A1=1 THEN 2970
3000 RESTORE
3010 FOR E2=DC TO DC+17:READ A$
3020 POKE E2,VAL("&H"+A$):NEXT E2:A1=1:GOTO 2970
3030 POKE HC(E1),FNA(DC+18,3):POKE HC(E1)+1,FNA(DC+18,1)
3040 IF A2=1 THEN 2970
3050 RESTORE 3090
3060 FOR E2=DC+18 TO DC+101:READ A$
3070 POKE E2,VAL("&H"+A$):NEXT E2:A2=1:GOTO 2970
3080 DATA E5,C1,21,0,0,3E,10,29,EB,29,EB,30,1,9,3D,C8,18,F5
3090 DATA 1,0,0,D5,7C,17,38,34,D1,E5,7A,17,38,38,E1,C5,6,0,3E,10,29,E5,C5,E1,ED,
4A,E5,ED,52,38,31,C1,E5,C1,E1,23,3D,20,ED,D1,3E,1,A2,C8,E5,21,0,0
3100 DATA ED,42,E5,C1,E1,EB,21,0,0,ED,52,C9,4,EB,21,0,0,AF,ED,52,18,C2,4,21,0,0,
AF,ED,52,EB,18,BE,C1,E1,18,D0
3110 REM < JMPｱﾄﾞﾚｽｶｷｺﾐ >
3120 IF LB=1 THEN RETURN
3130 FOR E1=1 TO LB-1
3140 IF HB(E1)=0 THEN 3190
3150 IF PEEK(HB(E1)+1)=&HFF THEN 3200
3160 E2=PEEK(HB(E1))
3170 IF HA(E2)=0 THEN PRINT "NOT no. ";E2:GOTO 3190
3180 POKE HB(E1),FNA(HA(E2),3):POKE HB(E1)+1,FNA(HA(E2),1)
3190 NEXT E1:RETURN
3200 R1=PEEK(HB(E1))
3210 FOR E2=R1+1 TO 200
3220 IF HA(E2)<>0 THEN 3180
3230 NEXT E2:PRINT "if jmp MISS!!":RETURN
3240 REM < ﾍﾝｽｳｱﾄﾞﾚｽ ﾓﾆﾀｰ >
3250 INPUT "ﾍﾝｽｳﾒｲ";VN$
3260 IF RIGHT$(VN$,1)=")"THEN 3290 ELSE G3$=VN$+"":GOSUB 2830
3270 IF EE=1 THEN EE=0:GOTO 310
3280 PRINT HEX$(HH):GOTO 310
3290 G6$=LEFT$(VN$,LEN(VN$)-1):GOSUB 2870:GOTO 3270
3300 REM < ｷﾞｮｳ ｱﾄﾞﾚｽ ﾓﾆﾀｰ >
3310 INPUT "ｷﾞｮｳ ﾊﾞﾝｺﾞｳ";E1
3320 IF HA(E1)=0 THEN PRINT "ｱﾄﾞﾚｽ ﾅｼ!!":GOTO 310
3330 PRINT HEX$(HA(E1)):GOTO 310
```

```
3340 REM < ｴﾝﾄﾞ ｼｮﾘ >
3350 GOSUB 2910:GOSUB 3110:DC=DC+A1*18+A2*102
3360 A1=0:A2=0
3370 E2=VAL("&H"+DD$):E1=DC-E2
3380 PRINT "ｼｮｳ ﾒﾓﾘ ";HEX$(VAL("&H"+DD$));" - ";HEX$(DC-1);"  ";E1;" ﾊﾞｲﾄ"
3390 PRINT "ｻﾌﾞﾙｰﾁﾝ ｱﾄﾞﾚｽ  ";ST$
3400 END
3410 PRINT "BA?? MISS":RESUME 3420
3420 DC=MX:LB=MB:LC=MC:LD=MD
3430 IF HI<251 THEN HA(HI)=0:PRINT Q$
3440 PRINT "ﾔﾘﾅｵｼ!!":GOTO 310
3450 PRINT "ｷﾞｮｳ ﾆｭｳﾘｮｸ MISS!!":GOTO 310
3460 REM < ｿｳﾆｭｳ >
3470 IP=1:GOTO 330
3480 FOR E2=HI+1 TO HZ
3490 IF E2>HI AND HA(E2)<>0 THEN M2=E2:GOTO 3510
3500 NEXT E2:GOTO 3450
3510 HA(HI)=HA(M2):M4=DC-1:DC=KZ
3520 GOTO 380
3530 M1=DC-KZ:R9=KZ-HA(M2)
3540 IF M1>100 THEN PRINT "ｿｳﾆｭｳ ｽﾍﾟｰｽ ｵｰﾊﾞｰ":GOTO 3420
3550 IF M4+M1>KM THEN PRINT "ﾒﾓﾘ ｵｰﾊﾞｰ":GOTO 3420
3560 FOR E2=M4 TO HA(M2) STEP-1
3570 POKE E2+M1,PEEK(E2)
3580 NEXT E2
3590 FOR E2=0 TO M1-1
3600 POKE HA(M2)+E2,PEEK(KZ+E2)
3610 NEXT E2
3620 FOR E2=M2 TO HZ
3630 IF HA(E2)<>0 THEN HA(E2)=HA(E2)+M1
3640 NEXT E2
3650 FOR E2=1 TO LB-1
3660 IF HB(E2)=0 THEN 3690
3670 IF HB(E2)>KZ THEN HB(E2)=HB(E2)-R9:GOTO 3690
3680 IF HB(E2)>HA(HI) THEN HB(E2)=HB(E2)+M1
3690 NEXT E2
3700 FOR E2=1 TO LC-1
3710 IF HC(E2)=0 THEN 3740
3720 IF HC(E2)>KZ THEN HC(E2)=HC(E2)-R9:GOTO 3740
3730 IF HC(E2)>HA(HI) THEN HC(E2)=HC(E2)+M1
3740 NEXT E2
3750 IF LD=1 THEN 3800
3760 FOR E2=1 TO LD-1
3770 IF HD(E2)>KZ THEN HD(E2)=HD(E2)-R9:GOTO 3790
3780 IF HD(E2)>HA(HI) THEN HD(E2)=HD(E2)+M1
3790 NEXT E2
3800 DC=M4+M1+1:IP=0:GOTO 300
3810 REM < ｻｸｼﾞｮ >
3820 INPUT "ｷﾞｮｳ";HI
3830 IF HA(HI)=0 THEN 3450
3840 FOR E2=HI+1 TO HZ
3850 IF HA(E2)<>0 THEN M2=E2:GOTO 3870
3860 NEXT E2:GOTO 3450
3870 LPRINT Y$;HI:M1=HA(M2)-HA(HI):M4=DC-HA(M2):R9=HA(M2)
3880 FOR E2=0 TO M4
3890 POKE HA(HI)+E2,PEEK(HA(M2)+E2)
3900 NEXT E2
3910 FOR E2=M2 TO HZ
3920 IF HA(E2)<>0 THEN HA(E2)=HA(E2)-M1
3930 NEXT E2
3940 FOR E2=1 TO LB-1
3950 IF HB(E2)=0 THEN 3980
3960 IF HB(E2)>HA(HI) AND HB(E2)<R9 THEN HB(E2)=0:GOTO 3980
3970 IF HB(E2)>R9 THEN HB(E2)=HB(E2)-M1
3980 NEXT E2
3990 FOR E2=1 TO LC-1
4000 IF HC(E2)=0 THEN 4030
4010 IF HC(E2)>HA(HI) AND HC(E2)<R9 THEN HC(E2)=0:GOTO 4030
4020 IF HC(E2)>R9 THEN HC(E2)=HC(E2)-M1
4030 NEXT E2
4040 IF LD=1 THEN 4080
4050 FOR E2=1 TO LD-1
4060 IF HD(E2)>R9 THEN HD(E2)=HD(E2)-M1
4070 NEXT E2
4080 HA(HI)=0:DC=DC-M1:GOTO 300
4090 REM < ﾊｲﾚﾂ ﾉ ｼｮﾘ >
4095 SI=0:SS=0:SL=0
4100 G6$=HJ$(TG)+"":GOSUB 2870:R2=HH:RY=FC(E1)
4110 RZ=INSTR(HG$(TG),",")
4120 IF RZ<>0 THEN SI=1:RZ$=LEFT$(HG$(TG),RZ-1):HG$(TG)=RIGHT$(HG$(TG),LEN(HG$(T
G))-RZ)
4130 IF SI=1 AND RZ$<":" THEN RZ=VAL(RZ$):SL=1 ELSE IF SI=1 THEN G3$=RZ$+"":GOSU
B 2830:RZ=HH
4140 IF HG$(TG)<":" THEN R1=VAL(HG$(TG)):SS=1
4150 IF II=1 THEN POKE DC,&HE5:DC=DC+1
4160 IF SS=1 THEN POKE DC,&H21:POKE DC+1,FNA(R2+R1*2,3):POKE DC+2,FNA(R2+R1*2,1)
:DC=DC+3:GOTO 4220
4170 G3$=HG$(TG)+"":GOSUB 2830:R1=HH
4180 POKE DC,&H2A:POKE DC+1,FNA(R1,3):POKE DC+2,FNA(R1,1)
4190 POKE DC+3,&H29
4200 POKE DC+4,&H11:POKE DC+5,FNA(R2,3):POKE DC+6,FNA(R2,1)
```

BASICコンパイラ　プログラム・リスト

```
4210 POKE DC+7,&H19:DC=DC+8
4220 IF SS=1 AND SL=1 THEN 4280
4230 IF SL=1 THEN DC=DC-3:POKE DC,FNA((2*RZ*RY+R2),3):POKE DC+1,FNA((2*RZ*RY+R2)
,1):POKE DC+2,&H19:DC=DC+3:GOTO 4270
4240 IF SI=1 THEN POKE DC,&HE5
4250 IF SI=1 AND SL=0 THEN POKE DC+1,&H2A:POKE DC+2,FNA(RZ,3):POKE DC+3,FNA(RZ,1
) ELSE IF SI=1 THEN POKE DC+1,&H21:POKE DC+2,RZMOD256:POKE DC+3,RZ¥256
4260 IF SI=1 THEN POKE DC+4,&H29:POKE DC+5,&H11:POKE DC+6,RYMOD256:POKE DC+7,RY¥
256:POKE DC+8,&HCD:POKE DC+9,1:POKE DC+10,0:POKE DC+11,&HD1:POKE DC+12,&H19:HC(L
C)=DC+9:DC=DC+13:LC=LC+1
4270 RETURN
4280 DC=DC-2:POKE DC,FNA(2*(RY*RZ+R1)+R2,3):POKE DC+1,FNA(2*(RY*RZ+R1)+R2,1):DC=
DC+2:GOTO 4270
4290 REM < ｴﾝｻﾞﾝｼﾞｭﾝｼﾞｮ >
4300 ERASE Z$:DIM Z$(20):TX=0:TY=0
4310 IF LEFT$(Z9$,1)="-" THEN Z9$="0"+Z9$
4320 LL=LEN(Z9$)
4330 FOR TT=1 TO LL
4340 Z1$=MID$(Z9$,TT,1)
4350 IF Z1$="(" THEN GOSUB 4450
4360 IF CC=1 THEN CC=0:GOTO 4320
4370 NEXT TT
4380 FOR TT=1 TO LL
4390 Z1$=MID$(Z9$,TT,1)
4400 IF Z1$>")" AND Z1$<"0" THEN TY=TY+1
4410 NEXT TT
4420 IF TY>1 THEN Z5$=Z9$+"":GOSUB 4580:Z9$=Z5$+"":RETURN
4430 IF TY=0 THEN Z$(TX)=Z9$+"":RETURN
4440 Z5$=Z9$+"":L1=LL:GOSUB 4790:Z9$=Z5$+"":RETURN
4450 FOR T6=TT+1 TO LL
4460 Z1$=MID$(Z9$,T6,1)
4470 IF Z1$="(" THEN RETURN
4480 IF Z1$=")" THEN Z5$=MID$(Z9$,TT+1,T6-TT-1):GOSUB 4580:GOSUB 4510:CC=1:RETUR
N
4490 NEXT T6
4500 RETURN
4510 FOR T7=TT TO T6
4520 MID$(Z9$,T7,1)=" "
4530 NEXT T7
4540 IF C0=1 THEN TX=TX-1:C0=0
4550 X$="●"+MID$(STR$(10+TX),2,2):MID$(Z9$,TT,3)=X$
4560 TX=TX+1:L1=LL:Z5$=Z9$+"":GOSUB 4750:Z9$=Z5$+""
4570 RETURN
4580 REM < Z5$ >
4590 L1=LEN(Z5$)
4600 FOR T1=1 TO L1
4610 Z1$=MID$(Z5$,T1,1)
4620 IF Z1$="*" OR Z1$="/" THEN GOSUB 4640:GOSUB 4750:GOTO 4590
4630 NEXT T1:GOSUB 4790:RETURN
4640 FOR E2=T1-1 TO 1 STEP-1
4650 Z2$=MID$(Z5$,E2,1)
4660 IF Z2$>")" AND Z2$<"0" THEN Z2$=MID$(Z5$,E2+1,T1-E2-1):GOTO 4680
4670 NEXT E2:Z2$=LEFT$(Z5$,T1-1)
4680 FOR E3=T1+1 TO LEN(Z5$)
4690 Z3$=MID$(Z5$,E3,1)
4700 IF Z3$>")" AND Z3$<"0" THEN Z3$=MID$(Z5$,T1+1,E3-T1-1):GOTO 4720
4710 NEXT E3:Z3$=RIGHT$(Z5$,LEN(Z5$)-T1)
4720 L5=E3-E2-1:MID$(Z5$,E2+1,L5)=SPACE$(L5)
4730 Z$(TX)=Z2$+"."+Z3$+Z1$:X$="●"+MID$(STR$(10+TX),2,2)
4740 MID$(Z5$,E2+1,3)=X$:TX=TX+1:RETURN
4750 Y2$="":FOR T2=1 TO L1
4760 Z1$=MID$(Z5$,T2,1)
4770 IF Z1$<>" " THEN Y2$=Y2$+Z1$
4780 NEXT T2:Z5$=Y2$+"":RETURN
4790 FOR T3=1 TO L1
4800 Z1$=MID$(Z5$,T3,1)
4810 IF Z1$>")" AND Z1$<"0" THEN Z2$=MID$(Z5$,1,T3-1):GOTO 4840
4820 NEXT T3
4830 C0=1:RETURN
4840 Z3$="":T5=L1
4850 FOR T4=L1 TO T3 STEP-1
4860 Z1$=MID$(Z5$,T4,1)
4870 IF Z1$>")" AND Z1$<"0" THEN GOSUB 4910
4880 NEXT T4
4890 Z$(TX)=Z2$+"."+Z3$
4900 RETURN
4910 Z4$=MID$(Z5$,T4+1,T5-T4):T5=T4-1
4920 Z3$=Z4$+Z1$+Z3$:RETURN
4930 REM < ﾆｭｳﾘｮｸ >
4940 PRINT CHR$(12): GOSUB 5050:LOCATE 0,4:PRINT STRING$(40,&H95):LOCATE 0,11:PR
INT STRING$(40,&H95):POKE &HF5A6,32
4950 LOCATE 10,2:PRINT "ｷﾞｮｳﾊﾞﾝｺﾞｳ";HI;:LOCATE 26,2
4960 IF IP=0 THEN PRINT "ｱﾄﾞﾚｽ ";HEX$(DC); ELSE PRINT "ｿｳﾆｭｳ";
4970 YZ$="":LOCATE 0,5:LINEINPUT H$:Z1=LEN(H$):Q$=""
4980 IF Z1>=39 THEN Z1$=CHR$(PEEK(&HF5A6)):H$=LEFT$(H$,39)+Z1$+RIGHT$(H$,Z1-39)
4990 Q$=H$+"":PRINT Q$:INPUT " OK (y or n)";S$
5000 IF S$="n" THEN 4930
5010 LPRINT Y$;HI;" ";Q$
5020 FOR E2=1 TO LEN(H$):Z1$=MID$(H$,E2,1)
5030 IF Z1$<>" " THEN YZ$=YZ$+Z1$
5040 NEXT E2:H$=YZ$+"":RETURN
5050 LOCATE 9,0:PRINT "**** BASIC ｺﾝﾊﾟｲﾗ ****":RETURN
```

機械語マスターの早道！

ハンド・コンパイル入門



●世田ヶ谷マイコンクラブ●
丹曽久山門

『BASICまでは何とかわかるようになったけれど，機械語はわからない』というのは良く聞く言葉で，筆者自身も半年前まで同じ実感を持っていました．

ところが，友人の角沢氏から，H・C――『ハンド・コンパイル』技法を伝受されてから，機械語のなんたるかがわかったような気がしてきたのです．そこで，1人でも多くの方にこの“秘法”を伝受していただきたいと思い，彼に再三本誌に投稿するようにすすめたのですが，PASCALコンパイラの開発に熱中している彼は，とても引き受けてくれないうえ，ついには，『君が書けよ』という始末です．そこで，雄を鼓して実力もかえり見ず，筆を取った次第です．

なお，使用機種は手持ちの関係で，PCにしましたが，H・Cの考え方は，Z80はもちろん，8080でも6800でも，真意さえ理解していただければCPU，MPUや機種に関係なく利用できます．

1.『ハンド・コンパイル』とは

読んで字のごとく，BASICで書かれたプログラムを腕力でオブジェクト――機械語プログラムにしようというものです．もちろん，BASICコンパイラがあれば問題ないのですが，我々が簡単に使えるBASICコンパイラがありませんから，腕力に頼らざるを得ないわけです．

結局，理解し易いBASICでシナリオを書いて，これを機械語に変換しようというものです．

2.ニモニックとマシン・コード

機械語を扱う以上，マシン・コードの理解が必要ですが，マシン・コード自身は人間には理解しにくいため，一般にニモニックを使います．しかし，このニモニックも，

❶CPU，MPUで異なる．
❷同じ80系でも，8080とZ80では，違う部分がある．
❸特に8080のニモニックは，理解しにくい．
❹ニモニックは，アセンブル用言語で，アセンブル時以外には，特に必要ない．

というわけで，**表1**のようなZ80と8080共通の主要マシン・コード表を作りました．どんな命令があるか理解してください．

記号は筆者の『独断と偏見』でつけましたが，マシン語をちょっとかじれば，理解していただけるものと思います．

英語でもその基本となるのは単語で，機械語の場合も命令の理解がないと，無駄なプログラムになったり，ついにはコンパイル不能の事態になりかねません．命令の意味を充分勉んでいただきたいと思います．

3.LET文

代入文です．LET文をコンパイルするとき，考えなければならない問題点は，

❶変数用メモリのアドレス配分
❷実数か変数か．
❸1バイトか，2バイトか．
❹10進数か，16進数か．
❺システム――特にV-RAMなど，I/Oに関係するデータの変更

などがあります．

BASICでは，変数アドレスは**DIM**文で領域さえ確保すれば自動的に配分し，出し入れも自動的にやってくれますが，機械語ではすべて人間がやらなければなりません．

まず，データの必要バイト数に応じて，**表2**のように変数アドレスを配分します．

特に，BASICの他に機械語プログワムが付いている場合は，すでに決まっている場合もあります．H・Cに入る前にはフローチャートの1つも書いて，プログラムの内容を充分理解しておく必要があります．

```
10  LET  M=3
20  LET  N=P
30  LET  X=32393
40  LET  Y=2
```

10は1バイトの実数，20は1バイトの変数，30は2バイトの実数，40は2バイトの変数の代入になります．

メモリには直接データを入れられないので，レジスタを使います．このとき，1バイト・データの場合は主としてAレジスタを，2バイトのデータの場合は，HLなどのペ

表1　80系主要命令一覧表

	A	B	C	D	E	H	L	M	I	Acc	,	C		
A→	7F	47	4F	57	5F	67	6F	77△		i→A		3E	DAA	27
B→	78	40	48	50	58	60	68	70△		nn→A		3A	$\bar{A}$	2F
C→	79	41	49	51	59	61	69	71△		M→A		7E	$\bar{n}$	ED44
D→	7A	42	4A	52	5A	62	6A	72△		DE→A		1A	C=1	37
E→	7B	43	4B	53	5B	63	6B	73△		BC→A		0A	$\bar{c}$	3F
H→	7C	44	4C	54	5C	64	6C	74△						
L→	7D	45	4D	55	5D	65	6D	75△		A→nn		32		
M→	7E△	46△	4E△	56△	5E△	66△	6E△	76		A→M		77		
I→	3E	06	0E	16	1E	26	2E	36		A→DE		12	RLC	07
A+	87	80	81	82	83	84	85	86△	C6	A→BC		02	RRC	0F
A+C+	8F	88	89	8A	8B	8C	8D	8E△	CE				RAL	17
A−	97	90	91	92	93	94	95	96△	D6				RAR	1F
A−B−	9F	98	99	9A	9B	9C	9D	9E△	DE	IN		DB		
A∧	A7	A0	A1	A2	A3	A4	A5	A6△	E6	OUT		D3		
A∀	AF	A8	A9	AA	AB	AC	AD	AE△	EE	DI		F3		
A∨	B7	B0	B1	B2	B3	B4	B5	B6△	F6	EI		FB	HLT	76
A:	BF	B8	B9	BA	BB	BC	BD	BE△	FE				NOP	00
	A	B	C	D	E	H	L	M	I	HL	DE	BC	SP	PC
+1	3C	04	0C	14	1C	24	2C	34△		23○	13	03	33	
−1	3D	05	0D	15	1D	25	2D	35△		2B○	1B	0B	3B	
●DJNZ		●10		JMP	Br	CALL	RET	i→		21○	11	01	31	
●AF↔AF′		08	→	C3	●18	CD	C9	→nn		22○	●ED53	●ED43	●ED73	
●r↔r′		D9	NZ	C2	●20	C4	C0	nn→		2A○	●ED5B	●ED4B	●ED7B	
			Z	CA	●28	CC	C8	HL→					F9○	E9○
			NC	D2	●30	D4	D0	HL↔			EB		E3○	
			C	DA	●38	DC	D8	HL+		29○	19○	09○	39○	
			PO	E2		E4	E0	HL+C+		●ED6A	●ED5A	●ED4A	●ED7A	
			PE	EA		EC	E8	HL−B−		●ED62	●ED52	●ED42	●ED72	
			P	F2		F4	F0	PUSH		E5○	D5	C5	F5	
			M	FA		FC	F8	POP		E1○	D1	C1	F1	

●はZ80のみの命令，○はコードの前にDD(X)，FD(Y)をつけてXレジ，Yレジとして使用，△は相対番地付き．

ア・レジスタを使います．

また，原則として機械語では10進数は扱えないので16進数に変換しなければなりません．40の３２３９３はBSのV-RAMの相当アドレスに変換しなければなりません．画面サイズもＢＳが32×16に対し，ＰＣは，40×25なので，この点も考慮する必要があります．

ＰＣでは，**?HEX$(n)** で，ｎに10進数を入れるとダイレクト・モードで16進数に変換されます．また，逆の場合は**?VAL("8Hh")** でｈに16進数を入れると10進数に変換されます．

また，ＰＣのV-RAMアドレスとカーソル位置の変換表を**表3**に示します．

すべてのアドレスを示すと莫大なものになってしまうので，5個間隔としてありますから適宜加減してください(なお，ＰＣは40桁の場合，１桁当り２バイトずつ加減する必要があります)．

```
L10    $03→A→C080          C200   3E033280C0
```

実数の代入には，イミディエイト・アドレッシングで，＄は６８で使われている実数記号です．

```
L20    C082→A→C081       C205   3A82C0   3281C0
L30   $FB3C→HL→C084    C20B   213CF8    2284C0
```

３２３９３はカーソル９，８に相当し，これはＰＣの11.10にだいたい相当します．

```
L40   $C088→HL→C086       C211   2A88C0   2286C0
```

なお，ＬＥＴのうち，Ｐ＝０のような変数のクリアがあります．０もイミディエイトで，＄00→Ａ（3E00）でよいわけですが，最も便利なA∀A（ＥＸ-ＯＲ Ａ）を使います．つまり，

```
50   LET   P=0 は
     A∀A→C082       C217   AF   3282C0
```

となります．

２バイト変数のクリアは実数と同じように，＄0000をペア・レジスタにロードします．



表２　変数アドレス配分例

変数名	バイト数	アドレス
M	1	C080
N	1	C081
P	1	C082
Q	1	C083
X	2	C084，5
Y	2	C086，7
Z	2	C088，9

表3　PC-8001カーソル↔V-RAMアドレス換算表

	0	5	10	15	20	25	30	35	39
0	F300	F30A	F314	F31E	F328	F332	F33C	F346	F34E
1	F378	F382	F38C	F396	F3A0	F3AA	F3B4	F3BE	F3C6
2	F3F0	F3FA	F404	F40E	F418	F422	F42C	F436	F43E
3	F468	F472	F47C	F486	F490	F49A	F4A4	F4AE	F4B6
4	F4E0	F4EA	F4F4	F4FE	F508	F512	F51C	F526	F52E
5	F558	F562	F56C	F576	F580	F58A	F594	F59E	F5A6
6	F5D0	F5DA	F5E4	F5EE	F5F8	F602	F60C	F616	F61E
7	F648	F652	F65C	F666	F670	F67A	F684	F68E	F696
8	F6C0	F6CA	F6D4	F6DE	F6E8	F6F2	F6FC	F706	F70E
9	F738	F742	F74C	F756	F760	F76A	F774	F77E	F786
10	F7B0	F7BA	F7C4	F7CE	F7D8	F7E2	F7EC	F7F6	F7FE
11	F828	F832	F83C	F846	F850	F85A	F864	F86E	F876
12	F8A0	F8AA	F8B4	F8BE	F8C8	F8D2	F8DC	F8E6	F8EE
13	F918	F922	F92C	F936	F940	F94A	F954	F95E	F966
14	F990	F99A	F9A4	E9AE	F9B8	F9C2	F9CC	F9D6	F9DE
15	FA08	FA12	FA1C	FA26	FA30	FA3A	FA44	FA4E	FA56
16	FA80	FA8A	FA94	FA9E	FAA8	FAB2	FABC	FAC6	FACE
17	FAF8	FB02	FB0C	FB16	FB20	FB2A	FB34	FB3E	FB46
18	FB70	FB7A	FB84	FB8E	FB98	FBA2	FBAC	FBB6	FBBE
19	FBE8	FBF2	FBFC	FC06	FC10	FC1A	FC24	FC2E	FC36
20	FC60	FC6A	FC74	FC7E	FC88	FC92	FC9C	FCA6	FCAE
21	FCD8	FCE2	FCEC	FCF6	FD00	FD0A	FD14	FD1E	FD26
22	FD50	FD5A	FD64	FD6E	FD78	FD82	FD8C	FD96	FD9E
23	FDC8	FDD2	FDDC	FDE6	FDF0	FDFA	FE04	FE0E	FE16
24	FE40	FE4A	FE54	FE5E	FE68	FE72	FE7C	FE86	FE8E

4.POKE, PEEK

POKE は，**LET** と同じです．**PEEK** は使う必要はまずありませんが，考え方としては**LET**と変わりません．

5.四則演算

代入ができるようになったら次は四則演算ですが，四則演算の詳細を書き初めると，１冊の本になってしまう上，類書やＮＥＣのプログラム・ライブラリなどに，詳しく述べられているので，ゲームで利用する程度のものについて説明しましょう．四則演算も代入と同じく，メモリで直接できないので，１度ＡレジスタやＨＬレジスタに代入してから行ないます．結果のストアも，代入文と変わりません．

❶　1バイトの加算

●実数との加算　　A＋i
●レジスタとの加算　　A＋r
●HLで示される
　メモリとの加算　　A＋M

をそれぞれ利用します．

❷　1バイトの減算

同じように，A-i，A-r，A-Mを使います．

❸　2バイトの加算

ＨＬ＋ＤＥなどのペア・レジスタ同士の加減などしかないので，データをこれらに代入してから行ないます．

❹　2バイトの減算

Ｚ80になって便利になった命令の１つに，HL-DE-Cなどの16ビットSUB（16ビットの減算）があるので，これを利用します．

ＢＳでは，Ｆ７ＣＣＨの16ビットＳＵＢサブルーチンを利用するか，補数の加算を利用します．

❺　掛け算，割り算

本格的なサブルーチンを作らなければなりませんが，ゲーム程度の数学では，シフトやローテイトを活用します．

つまり，『ビットを１つ左にずらす』ということは，２倍すると同じことなので，精度がラフなゲームでは，10倍するようなところでは，**SLA**(CB27) を３回重ねて８倍で誤魔化してしまうというような芸当で，プログラムを簡略してしまえるわけです．

逆に右にずらすと 1/2 になるので，1/5のようなところは**SRL**(CB３F) を２回使って簡略化できます．

❻　CURSOR, PRINT, CLEAR

PRINT 文にはメッセージとデータの表示があり，**CURSOR** と関連させたサブルーチンを準備してあるので（後述），これを利用してください．

CLEAR はＰＣの０４５ＡＨにサブルーチンがありますが，簡単な利用法については後述します．

❼　GOTO, GOSUB, RETURN

行番号を飛び先アドレスに換え，**JMP**（Ｃ３または18），**CALL**(ＣＤ)，**RET**(Ｃ９）に置き換えるだけです．

❽　IF-THEN

IF文には

❶変数と実数の比較による分岐
❷変数と変数の比較による分岐

があります．

一方，機械語の比較命令には，

❶Aレジスタとイミディエイト・データ
❷Aレジスタと M (HLで示されるメモリ)
❸Aレジスタとレジスタ
があるので，適宜使い分けます．また，条件判断には，

❶ ＝ なら	JZ (CAまたは28)
❷ ＜＞なら	JNZ(C2 または20)
❸ ＞＝なら	JP(F2)
❹ ＜ なら	JM(FA)

があります．

【例2】
```
 50 IF  M=5     THEN  100
 60 IF  N<>P  THEN  110
 70 IF  X>=Y  THEN  120

L50  C080→A : $05        C250  3A80C0  FE05
          JZ  L100       C255  CA  _ _

L60  C082→HL             C260  2182C0
     C081→A : M          C263  3A81C0  BE
          JNZ  L110      C267
                               C2 _ _
L70  C086→DE             C270  ED5B86C0
     C084→HL-DE          C274  2A84C0  ED52
          JP  L120       C279  F2 _ _
```

なお，＞と＜＝は代入するレジスタや飛び先を逆にすればよいわけです．

❾ FOR-NEXT-STEP

FOR-NEXT文には，

❶ FORとNEXTの間に仕事の入らないTIMERとかWAITとかDELETとか呼ばれる時間待ちだけのもので，機械語ではスピードが早すぎて困る場合に多用するので，サブルーチンにしてありますからそちらを見てください．

❷ 一般のFOR-NEXT文で，カウンタを設け，これをSTEP分だけ増減させ，IF文で条件判断して分岐すればよいわけです．

【例3】
```
 80 FOR  I=1  TO 7  STEP  2
 81 PRINT
 82 NEXT  I

L80   $06→B             C280  3E06
      (PRINT)           C282  CD  _ _
         B-2            C285  05  05
         DJNZ  L81      C287  10F9
```

07になっているのを06にしたのは，条件判断は0で行なうのが簡単だからです．

```
 90 FOR  R=-3326  TO  -3252  STEP  10
 91 POKE R, 233
 92 NEXT  R
```

RはV-RAMのアドレスを指すので，システムに応じて変えます．

つまり，

```
   -3326=CU. TOP+1, 1   =F301
   -3252=CU. END-1, 1   =F34C
```

なので，F301をHL，F34CをDEに入れ，LETで♡マークをV-RAMに代入したのち，Lを＋0Aしながら，HL－DEを行ない，IF文で条件判断し，HL－DE＝0になるまで繰り返せばよいことになります．

❿ RND, RANDOMIZE

RNDはサブルーチンを見てください．

RANDOMIZEはスタートしてから最初にキーインされるまでRND変数を更新することによって行なっています．

⓫ GET

PCのキーコードは，表4のように，I/O No.とコードで決まります．たとえば，テンキーの1と3でパドルを左右に動かすようなときは，

```
      IN.0→A : ①              DB00   FEFD
      JNZ   L1                2002
      (LEFT)                  2B
L1           A : ③      L01   FEF7
      JNZ   L2                2002
      (RIGHT)                 23
L2                      L02   ———
```

ASCIIキーを，ゲームでたたかれてはたまりませんので，写真のように，シーメンス・スイッチやマイクロスイッチを，テンキーとパラレルに配線したパドルを自作してあります．

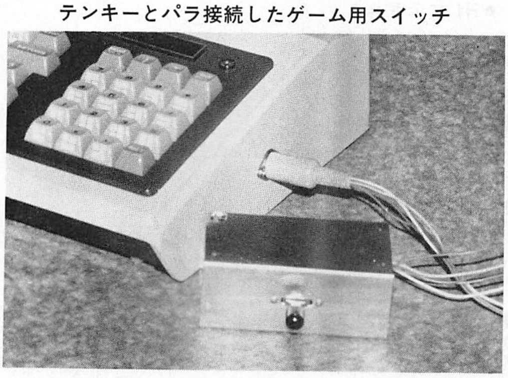


表4　キーのI/O No.とコード一覧表

I/O No.	FE	FD	FB	F7	EF	DF	BF	7F
0	0	1	2	3	4	5	6	7
1	8	9	*	+	=	,	.	RET
2	@	A	B	C	D	E	F	G
3	H	I	J	K	L	M	N	O
4	P	Q	R	S	T	U	V	W
5	X	Y	Z	[	]	¥	∧	—
6	0	1	2	3	4	5	6	7
7	8	9	:	;	.	,	／	
8	HOME	↑	→	INS	GRA	カナ	SHIFT	CNT
9		f・1	f・2	f・3	f・4	f・5	SP	ESC

（表の「コード」は FE〜7F の各列にまたがる見出し）

テンキーとパラ接続したゲーム用スイッチ



⑫ INIT と SBR

以上で，一般のゲーム位でしたら，簡単に機械語にコンパイルできることがわかったと思います．最後に，機械語プログラムを作る上での２～３の注意と汎用サブルーチンについて説明します．

1)　ORG：スタート・アドレスです．特にどこでなければならないというルールはありませんが，ＰＣでは，Ｃ100やＤ000にすれば良いでしょう．

2)　SPの決定：オブジェクトには，スタック・ポインタを初めに決めなければなりません．ＰＣが＋１されるのに対し，ＳＰは－１されるので，メモリ・エンド＋１にするか，ORGの前に，メモリの余裕があるときはORGを指定すれば良いでしょう．

3)　汎用SBR：先に説明したようなサブルーチンをORG付近に置けば，次回プログラムを作るとき，前のオブジェクトを利用することによって，キーインが省けて便利です．

あとがき

いかがだったでしょう．はじめて機械語に接した方にはもち論，チンプンカンプンだったでしょうが，多少機械語をかじったことのある方でしたら，『ナーンダ』と思われた方も多かったのではないでしょうか！

そうなんです，ＢＡＳＩＣも機械語もやっていることは同じですから，簡単に翻訳できるわけです．

次回には，I/O 誌に掲載されたプログラムの内，評判の高かったものをお手本としてご紹介する予定ですが，皆さんも一つ挑戦してみてください．

New Products

カセット・ベースの BASICコンパイラ

ハドソンソフト　中本伸一

ハドソンソフトでは現在BASICコンパイラを計画中です．私たちが最近発表したFORMは，整数型言語のコンパイラでその構造も小規模にまとまっていますが，完全自立型のオブジェクトを生成するために多くの方々から反響がありました．

今回計画中のBASICコンパイラは基本的に外部ファイルをカセットとし，整数型からさらに一歩進め，ディスク・ベースの実数型コンパイラとの中間に位置するコンパイラで，完全自立型のオブジェクトを生成します．

外部ファイルはカセット・レコーダを使用するという条件を付けたので，フロッピーなどの周辺機器を一切使用しないためカセットのかけ換えが多くなるのは避けられません．

ともあれ，48Ｋバイトの RAMを持つMZ-80K/Cだけあれば，BASICテキストをコンパイルできるコンパイラですから，MZユーザーはほとんどこのBASICコンパイラを走らせることができることと思います．

■BASICコンパイラの仕様

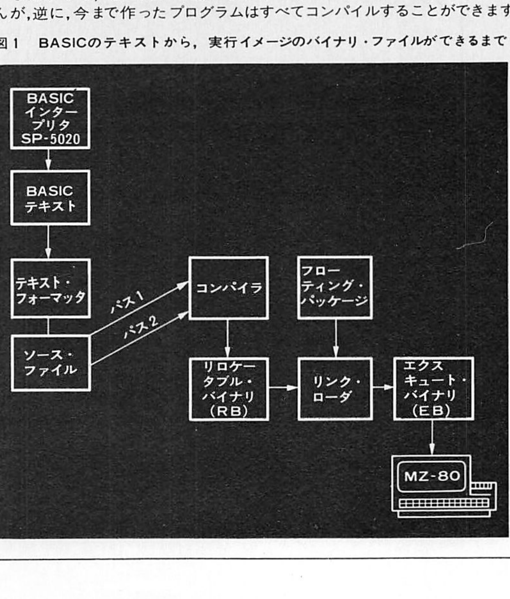
このBASICコンパイラの仕様が決まりましたので，紹介かたがたどういう構成になっているかを説明します．

BASICコンパイラでコンパイルするテキストは，シャープのBASIC SP-5020で作ったもので，この中間コードをコンパイルし，Z80のマシン・コードに展開します．

インタープリタでテキストを作ることにしたので，この時点で実行テストやシンタックス・チェックを行ない，完全に動作するプログラムを作ること必要条件となります．

したがって，SP-5020で動かないプログラムはコンパイルできませんが，逆に，今まで作ったプログラムはすべてコンパイルすることができます．

図１　BASICのテキストから，実行イメージのバイナリ・ファイルができるまで



このテキストをコンパイルするためには，それを，カセットに落した形で用意します．

BASICコンパイラはカセット4本のセットで発表の予定です．この４本は，

- ・テキスト・フォーマッタ　(TEXT FORMATTER)
- ・コンパイラ　(COMPILER)
- ・リンク・ローダ　(LINK-LOADER)
- ・フローティング・パッケージ　(FLOWTING PACKAGE)

であり，この他に必要なものは，SP-5020 BASICとブランク・カセット数本です．

これらのソフトを使用してコンパイルする訳ですが，その手順を以下簡単に述べます．

■コンパイルの手順

テキスト・フォーマッタをロードし，SP-5020で作ったBASICテキストを読み込ませます．このフォーマッタは，BASIC テキストをコンパイラが読めるようなファイルに，変更するためのもので，これによって出力されたファイルはコンパイラのソース・ファイルとなります．

これの働きは連続して記録されているテキストを256 バイトずつのブロックにわけることにより，テキスト全部をメモリに入れなくてもコンパイルできるようにするためです．

したがって，このフォーマッタは単にカセットのフォーマットを変換するだけの機能しか持ちません．

コンパイラは，テキスト・フォーマッタで作ったソース・ファイルを読み込み，これをマシン・コードに変換します．

このコンパイラは２パス型式で，ソース・ファイルを２回読み込ませます．パス１でマシン・コードに落とすのと同時に，サブルーチンや変数のテーブルを作り，パス２でアドレスを生成します．

メモリにあまり負担のかけない小さなプログラムであれば，一度にすべてソース・ファイルをメモリに読み込み，その中でパス１～２を行なうことができるので，パス２における読み込み時間は短縮できます．

こうしてメモリ中に，リロケータブルなオブジェクト・ファイルが作られるので，これをいったんカセットに落とします．

コンパイラによって出力されたリロケータブル・バイナリ（RB）をリンク・ローダによって任意のアドレスにリンクします．

コンパイラから出力されたRBは，サブルーチン・パッケージを持ってないので，RBをリンク後，パッケージの入っているカセットを読み込ませ，いっしょにリンクします．こうして実行イメージのバイナリ・ファイル（EB）が作られ，これをカセットへ出力します．

こうして作ったテープは，BASICテキストを完全にマシン語へ展開したプログラムが記録されているので，MZモニタからロードすれば，すぐに実行できます．

文章で書くと少し複雑に聞こえるかもしれませんが，図式化すると一目で流れがわかると思います（図１）．

■使用感，etc.

結局，このコンパイラはフロッピーディスクを使用しないために，時間のほとんどがテープの読み込みに消費されてしまいます．まるでシャープのアセンブラと同じような使用感であると思われます．

ともあれ，カセットを駆使したコンパイラは実験的に成功しているので，このBASICコンパイラは近いうちに発表できることと思います．

価格などはまだ決まっていませんが，一応８月に発表する予定です．ご期待ください．

BASICとマシン語が一度にロードできる！

BASIC マシン語 Auto Loader

PC-8001用オート・ローダ

大宮万岁

インベーダーやヘッド・オンなど，BASIC とマシン語の両方を使う場合，キーイン操作が面倒です．これを1回のキーインでロードし，さらに自動的に RUN できたのでお知らせします．

オート・ローダについて

普通，**cload**“——”でも L でもロード終了とともに必ずモニタに戻りキー入力待ちになり，人間が何か次の命令をキーインしなくてはなりません．ところが，PC-8001ではこのキーイン待ちになる前にファンクション・キーフラグが立っていると，キーポインタの指す番地からの文字列を“00”があるまで PC-8001自身が，キーインしてくれます(I/O '80年2月号 p.103の椚田様感謝)．

オート・ローダ・テープはファンクション・キーポインタの指す番地に命令の文字列を入れ，ファンクション・キーフラグを立てればいいのです．幸いなことにファンクション・・キーポインタ (EDC0, EDC1) はリセットで，EAC0 にセットされ，ファンクション・キーセット命令が使えます．そして，ファンクション・キーフラグ (EA86) の近くなので，テープも短くなります．

プログラム概要

100行～130行で次のように EAC0 からの命令を書き込みます．最初の EABC からの **run** CR と最後の方の(EAE0)←0 は f·5 キーを **run** CR に再生させるためです．

140行～150行はその都度キーインしなくてもいいように f·8 キーに入れました．実は f·8 キーを押してその内容の打ち出しが終わっていない場合，ファンクション・キーフラグが立ったままなのです．

使い方

リスト1のプログラムをキーインし，RUN してください．プログラム名をキーインすると，モニタに戻ります．ここで，普通やるように，目的の BASIC とマシン語をPC-8001に入れます．このとき，f·5 キーで試験的に RUN させたりするとせっかく入れた BASIC プログラムが消えるので注意してください．

次にテープ・レコーダを録音状態にし， f·8 キーを押してください．オート・ローダが書き込まれます．後は普

写真1　オート・ローダ　プログラムを実行させたところ

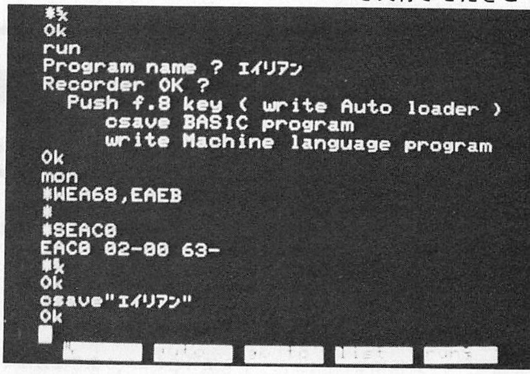


写真2　BASICのロードが終了すると，自動的にマシン語のロードに入る．

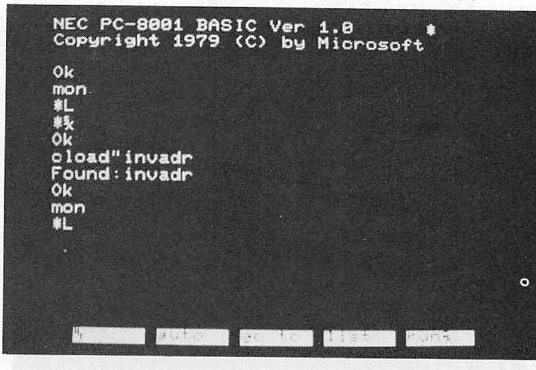


図1　100行～130行の利用

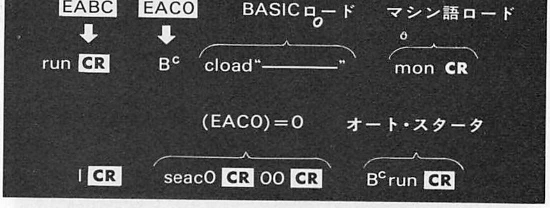
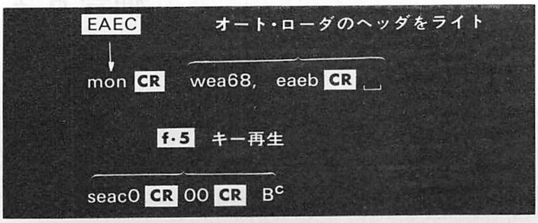


図2　140行～150行の作用



通にBASICプログラムをセーブし，次に，マシン語を書き込みます．BASICのプログラム名は先にキーインしたものと同じにしなければいけません．これでテープができあがりました．巻き戻して，Reset mon CR l CR してみてください．

最後に

f·6 キー～ f·7 キーまで破壊されてしまいます．もし，これらのキーを使う場合はBASICプログラムの最初に適当に付け加えてください．また，ファンクション・キーフラグとポインタを使うとI/O '80年7月号のアセンブラの最後から，BASICプログラムの**REM**文（'文）でソース・リストを付け加えることができます（**リスト2**）．訂正の都度，全部キーインする必要がなくなり，便利です．さらに，BASICプログラムの書き換えも人間がCRするのではなく，PC-8001に自動的にさせて，RUNさせることもできます．

リスト1　オート・ローダ プログラム・リスト

```
10 'Auto Loader
20 C$=CHR$(13):D$=CHR$(2):E$=CHR$(34)
30 INPUT"Program name ";N$:N$=LEFT$(N$,6):L=LEN(N$)+1
40 N$=N$+"
50 MID$(N$,L,1)=E$
60 PRINT"Recorder OK ?"
70 PRINT"  Push f.8 key ( write Auto loader )"
80 PRINT"        csave BASIC program"
90 PRINT"        write Machine language program"
100 KEY 5,"run"+C$+D$+"cload"+E$+N$
110 POKE &HEACB,ASC(MID$(N$,5,1))
120 KEY 6,MID$(N$,6,1)+C$+"mon"+C$+"l"+C$+"seac0"+C$+"0":POKE &HEADB,48
130 KEY 7,C$+D$+"run"+C$
140 KEY 8,"mon"+C$+"wea68,eaeb"+C$:POKE &HEAFB,32
150 KEY 9,"seac0"+C$+"00"+C$+D$
```

リスト2　ソース・リスト付きPC-ASMの追加訂正部分

```
110 CLEAR 700,&HCFFF:DEFINT A-N:DEFSTR O-W          ←①
390 UE="  44 45 46 47 4D 56 57 5E 67 6F ":Z=53248!:L=0:LL=0     ←②
1010 PRINT:PRINT "READY e-ENd l-LBL a-ASS d-DSA s-SRC ?":S=INPUT$(1):IF S="e"TH
EN END                                                ←③
1030 ON INSTR("lads",S)GOSUB 3420,4000,1050,8000←④
3035 LPRINT FNP(HEX$(XX));" ";TT;TAB(14);TN;TAB(21);S;TAB(26);TA;TB←⑤
3440 LPRINT"LABEL=";L:FOR I=0TOL-1:LPRINT TL(I);": ";HEX$(XL(I));"H",:NEXT:LPRIN
T:RETURN                                              ←⑥
4130 IF LL=0 THEN POKE AY-1,0:POKE &HEA68,0:WA="":RETURN ELSE W=TL(31-LL):X=XL(3
1-LL):GOSUB6060:IF ER=1 THEN PRINT"UNDF ";W,HEX$(X):GOTO 4170   ←⑦ ←⑧
5035 IF WA="s"THEN 7000 ELSE 4010←⑨
7000 POKE AY-1,0:IF AY>AZ THEN POKE &HEA68,0:WA="":RETURN←⑩
7010 AX=(AY+7)/AF+AF:POKE &HEDC1,AX:POK E &HEDC0,AY+7+(AF-AX)*AF:POKE &HEA60,1 }⑪
7020 AY=AS+(AT-AF)*AF
7030 AS=PEEK(AY):AT=PEEK(AY+1):POKE AY-1,13:GOTO 4010
8000 AF=256:AZ=PEEK(&HEFA0):AZ=AZ+(PEEK (&HEFA1)-AF)*AF-3
8010 WA="s":AY=VARPTR(WA):AY=PEEK(AY+1)+(PEEK(AY+2)-AF)*AF-8:AY=PEEK(AY)+(PEEK(A }⑫
Y+1)-AF)*AF:AS=PEEK(AY):AT=PEEK(AY+1):GOTO 7000
9000 'Print equ 52edh
9010 'Pc8001 equ 81h
9020 'loop ld hl,data
9030 ' call Print
9040 ' ld a,(count)
9050 ' dec a
9060 ' ld (count),a                                   }⑬
9070 ' jr nz,loop
9080 ' jp Pc8001
9090 'data dc 'Pc-8001 Assembler'
9100 ' db 32,32,0
9110 'count db 10
9120 ' end
```

①，② BASIC エリア拡張
③，④ ソース用
⑤，⑥ プリンタ
⑦ 9120のendの後に"00"を再現
⑧ f·キーフラグ0
⑨ ソースならソースアセンブラへ
⑩ 終わり
⑪ f·キーフラグとポインタ・ソース・リストに設定．
⑫ ソース・リストの先頭番地探し．
⑬ 最後にソース・リスト(ここでは9000～9120)を付けておき，1010行で S をキーインすると連続してアセンブルされる，endで戻る．

《注意》
ERRORやSTOPで途中止まった場合，右のようになります．これを救うには，壊れた行と次の行をCRT上に出し，⬇印位置へカーソルをセットし，CTRL E で元に戻します．これだけでは次の行が消えてしまうので，もう1度次の行を入れてください．

```
   ⬇
9070 'ww equ 0d414h14883x#:rr equ 0d418h
9080 'rr equ 0d418h
```

●変数，GOTO，GOSUB文の整理用に

仙台万岁

PC-8001 バリアブル・リスト

Variable List

BASICでプログラムを作るとき，どの変数を使ったか，使ってない変数はどれかなどと，よく迷うものです．一応動作すると，もう変数一覧表を改めて書き出すのも面倒で，そのまま忘却の彼方，ある日，バグや改良希望が出てきて大あわて，思い出すのにひと苦労ということがよくあるものです．またI/O誌などに発表されたプログラムをしても走らない，改造したいなど，変数一覧表があれば頭脳明晰，RUN一発OKです．

仕様

- ●80桁，40桁指定
- ●USR，FN，COLORもOK
- ●GOTO，GOSUBの飛び先きによる一覧表付き
- ●同一行での使用回数付き

使い方

CLEAR300，&HE400

をまず入力します．あとは，[G] E500 [CR] です(実行例1)．

width 80 ?

と聞かれたら，80桁にしたいときだけ [y]，40桁のときには [y] 以外のキーを押してください．

Printer ?

も同様です．

マージャン・ゲームのような長いプログラムのときは，10秒くらいかかりますが，普通のプログラムではすぐに打ち出してきます．

途中で止めたいときには [ESC] キー，再スタートも [ESC] キーです．

モニタに帰るときは [STOP] キーです．ただし，プリンタ出力時に [ESC] キーは効きません．

★COLOR文

カラーモニタを買えない人のために付けました．カラー使用のプログラムをモノクロ用に改造するとき，便利です．

また，BEEP音がウサギ小屋の騒音限度を越えるときには，0E554H，0E736H，0E79FHの0B5Hを0B2Hに，0E970Hからの43H，4FH，4CH，4FH，52H（COLOR）を，42H，45H，45H，50H，20H（BEEP␣）にします．マイコンは皆のものです．金のない人々は助け合いましょう．

★実行例

実行例1 使用方法CRTコピー

```
*L
*SEAC0
EAC0 73-00 65-
*
Ok
Variable List
clear 300,&he400
Ok
mon
*L
*
Ok
load BASIC p. & mon RET GE500 RET
cload"

mon
*GE500
Variable List for PC8001
  width 80 (y) ?

ESC- escape  STOP- Monitor

Printer (y) ?
*G124A
```

実行例1に示したとおりです．これは，

Printer (y) ?

のときに，[STOP] キーを押し，画面コピー [G] 124A [CR] をしたものです．実行例2はプログラムの実行結果です．

★その他

このプログラムによる変数リストは，BASICプログラムのフリーエリアに置かれます．BASICのワーク・エリアを破壊しないので，安心して使えます．

これを作りながら気付いたのですが，16進4桁を0F0A8Hと0F0A9Hに入れ，5桁の10進ASCIIに変換して，(HL)～(HL＋5)に入れるサブルーチンを使用

メモリ・マップ

アドレス	領域
8020 (D020)	BASIC プログラム
	BASIC ワークエリア
	バリアブル・リスト
	BASICスタック
E500	バリアブル・リスト・プログラム
E8FF	バリアブル・リスト・定数
E984	バリアブル・リスト・ワークエリア
E9E1	

するときには，Bレジスタの値に注意してください．不必要なコンマが打たれることが，あります．

このプログラムの構造はあまりキレイではないので，リロケートのために定数エリアを記します．要は変数一覧表が欲しいのですから．

実行例2　ギャラクシアン（I/O '80年10月号）

```
  AA      160   200   250   260   290
          300   370
  B       190c
  BF      260   270
  F       310b
  I       350
  COLOR   150   180   210   240   270
          280   320   380   390   400
          410   430   440   460   490
          500   510
  USR0    290b
  USR1    300b
  USR2    160b
  USR3    250b
  USR4    260b
  USR5    200b
  USR6    370b
 GOTO & GOSUB
   170    360   530
   320    310
   370    170
   450    450
   460    310
   530    530
```



注）プリンタ出力時に ESC キーは効きません．

◘参考文献

1）欄田鷹男：“Token List”, I/O，'80年2月号

Variable Listプログラム

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E500 ED 73 8A E9 31 00 E5 CD C5 E8 2A A4 EF 22 84 E9 :9AF
E510 3E FF 77 AF 32 8E E9 22 86 E9 2A 54 EB E5 E1 5E :92A
E520 23 56 CD D3 5E D2 F6 E6 D5 23 5E 23 56 EB 22 88 :889
E530 E9 EB 23 CD CA E5 B7 28 E5 FE 8F 28 E1 CD D8 E5 :B57
E540 D2 B5 E5 FE 22 47 28 4D FE 84 28 47 FE DB 28 26 :860
E550 32 8E E9 FE B5 28 36 47 E6 EC EE 0C 20 D5 78 FE :938
E560 0D 28 3D FE 0E 28 39 EB 21 76 E9 01 0D 00 ED B1 :5F6
E570 6E 26 00 19 18 BD 57 CD CA E5 D6 11 FE 0A 30 08 :67C
E580 C6 30 5F 7A 53 C3 14 E6 3E 20 2B 18 F5 11 00 00 :586
E590 C3 14 E6 06 3A 7E 23 B8 28 99 B7 CA 1E E5 18 F5 :7A8
E5A0 5E 23 56 23 E5 FE 0E 28 07 EB 23 23 23 5E 23 56 :545
E5B0 3E EC C3 15 E6 57 1E 20 06 00 CD CA E5 CD D1 E5 :882
E5C0 38 21 04 05 20 F4 5F 04 18 F0 7E 23 FE 20 C0 18 :578
E5D0 F9 FE 30 D8 FE 3A 38 07 FE 41 D8 FE 5B 30 02 A7 :8BF
E5E0 C9 37 C9 FE 24 28 10 FE 23 28 0E FE 21 28 0A FE :6C9
E5F0 25 28 06 06 20 18 06 3E 2A 47 CD CA E5 FE 28 28 :510

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E600 03 2B 18 02 CB F8 3A 8E E9 FE DC 20 06 CB F0 AF :826
E610 32 8E E9 78 E5 CD 1C E6 E1 C3 33 E5 2A 84 E9 BE :9E6
E620 28 21 38 05 CD C5 E6 18 F6 D5 01 0B 00 CD A3 E6 :743
E630 77 23 0E 09 71 23 70 23 D1 CD 8F E6 23 3E FF 77 :6C2
E640 23 77 C9 23 E5 23 23 7A BE 23 38 32 20 06 7B BE :5D5
E650 38 2C 28 05 CD C5 E6 18 EE 23 4E 23 46 54 5D 09 :5A3
E660 03 03 EB 70 2B 71 EB 01 02 00 CD A3 E6 CD 9A E6 :78E
E670 01 02 00 E1 5E 23 56 EB 09 EB 72 2B 73 C9 2B 01 :59F
E680 06 00 D5 CD A3 E6 D1 CD 8F E6 01 06 00 18 E4 72 :7B9
E690 23 73 23 01 03 00 71 23 70 23 EB 2A 88 E9 EB 73 :5C8
E6A0 23 72 C9 EB 2A 86 E9 B7 ED 52 23 E5 2A 86 E9 54 :8CD
E6B0 5D 09 22 86 E9 EB E5 2A 8A E9 CD D3 5E E1 38 0B :886
E6C0 C1 ED B8 23 C9 23 4E 23 46 09 C9 21 50 38 CD ED :761
E6D0 52 21 20 61 22 92 E9 21 74 20 22 94 E9 2A 88 E9 :680
E6E0 EB 21 96 E9 CD 57 E8 3E 0D 77 23 AF 77 21 92 E9 :83E
E6F0 CD ED 52 C3 9B E8 DD 2A 84 E9 AF 32 8E E9 DD 7E :A79

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E700 00 DD 23 FE FF CA 9B E8 CD 26 E7 DD 6E 00 DD 23 :96F
E710 DD 66 00 DD 23 11 FF FF CD D3 5E CA FE E6 CD 85 :A50
E720 E7 CD E0 E7 18 EF 21 20 20 DD 23 DD 23 FE EC 28 :8F5
E730 43 FE DB 28 22 FE B5 28 1E CB 77 C4 5E E7 CB 7F :8F4
E740 28 05 21 28 29 CB BF FE 2A 20 04 3E 24 18 08 FE :4F5
E750 20 20 04 7D 6C 26 20 32 8F E9 22 90 E9 C9 CB B7 :703
E760 F5 3A 8E E9 B7 20 0B 21 5B E9 CD 7A E7 3E FF 32 :88A
E770 8E E9 F1 C9 32 8F E9 21 60 E9 11 92 E9 01 11 00 :7E3
E780 ED B0 C3 83 E8 DD 4E 00 DD 23 DD 46 00 DD 23 03 :81C
E790 C5 FD E1 3A 8F E9 FE EC 28 3C FE DB 28 19 FE B5 :A70
E7A0 28 28 22 93 E9 2A 8F E9 3A 91 E9 22 95 E9 32 97 :7AD
E7B0 E9 3E 20 32 92 E9 C9 7C 32 96 E9 3E 55 32 93 E9 :82B
E7C0 21 53 52 22 94 E9 3E 20 18 E4 21 6F E9 11 92 E9 :6C4
E7D0 01 06 00 ED B0 C9 55 5C 21 92 E9 CD 57 E8 23 C9 :7B2
E7E0 11 FF FF 21 97 E9 E5 3E 20 DD 6E 00 DD 23 DD 66 :881
E7F0 00 DD 23 FD 2B FD 2B CD D3 5E 20 10 E3 BE 20 03 :742

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E800 3E 61 77 34 CD 40 E8 28 2D E3 18 DB 54 5D E3 23 :721
E810 01 92 E9 3A 8C E9 D6 05 47 7D 91 B8 38 13 E5 D5 :818
E820 CD 77 E8 D1 E1 CD 40 E8 38 02 E1 C9 CD 4A E8 18 :9CE
E830 0A CD 40 E8 38 05 CD 77 E8 E1 C9 CD 57 E8 18 C9 :8FF
E840 D5 11 FF FF FD 19 FD 23 D1 C9 21 92 E9 06 06 3E :89A
E850 20 77 23 05 20 FB C9 EB 22 A8 F0 EB E5 01 00 00 :719
E860 CD 9F 30 E3 06 04 0E 20 3E 30 BE 20 05 71 23 05 :4A1
E870 20 F8 E1 3E 20 77 C9 00 3E 0A 77 23 3E 0D 77 23 :55E
E880 3E 00 77 21 92 E9 CD ED 52 3A 8D E9 B7 C4 A2 E8 :912
E890 CD AC 0F FE 1B CC B0 E8 FE 03 C0 ED 7B 8A E9 C3 :A64
E8A0 3C 5C 32 49 EB 21 92 E9 CD ED 52 AF 32 49 EB C9 :884
E8B0 CD BD E8 CD 98 E8 CD AC 0F FE 1B 20 F6 CD AC 0F :9FE
E8C0 FE 1B 28 F9 C9 21 FF E8 CD ED 52 CD AC 0F B7 28 :97E
E8D0 FA FE 79 3E 50 28 02 3E 28 32 8C E9 CD A3 09 CD :77C
E8E0 AC 0F B7 20 FA 21 2E E9 CD ED 52 CD AC 0F B7 28 :837
E8F0 FA CD 98 E8 FE 79 3E 01,28 01 AF 32 8D E9 C9 56 :89C

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E900 61 72 69 61 62 6C 65 20 4C 69 73 74 20 66 6F 72 :5F3
E910 20 50 43 38 30 30 31 0A 0D 0A 20 20 77 69 64 74 :395
E920 68 20 38 30 20 28 79 29 20 3F 0A 0D 0A 00 45 53 :2F2
E930 43 2D 20 65 73 63 61 70 65 20 20 53 54 4F 50 2D :4B4
E940 20 4D 6F 6E 69 74 6F 72 0A 0A 0D 50 72 69 6E 74 :536
E950 65 72 20 28 79 29 20 3F 0A 0D 00 46 4E 0A 0D 00 :2E2
E960 47 4F 54 4F 20 26 20 47 4F 53 55 42 0A 0D 00 20 :356
E970 43 4F 4C 4F 52 00 1F 08 1D 04 1C 02 1E 01 FF 01 :304
E980 0C 02 0F 01 52 45 54 0D 2A 0D 00 C8 28 50 45 47 :3E9
```



PC-8001

サーチダンププログラム

■石井　晴正

このプログラムは，N-BASICを調べるために作ったサーチおよびダンプのプログラムです．N-BASICの中の文字列を調べ，その文字列をコールしている番地を調べます．

使い方

プログラムをRUNさせると，実行例1を表示するので，サーチ，ダンプのときは[1]を，ターミナル・キーONのときは[2]を入力します．キーONのときは，コントロール・コードの表示や画面をプリントすることができます．[f･6]でコントロール・コード表示のON-OFFを繰り返し，[f･9]で画面プリント，[f･8]でターミナル・キーモードをOFFできます．

[1]を入力するとBGIN，ENDアドレスを聞いてくるので，16進でアドレスを入力します．

次に，プリンタに出力する場合は[Y]を入力します．このプログラムでは，yes or noのとき[Y]だけで，判別しているので，[Y]以外はすべてnoと見なされます．

続いてサーチかダンプか聞いてきます．ダンプなら[S]以外のキーを押します．これでダンプされます(実行例2)．

サーチの場合[S]を入力します(実行例3)．

以下，サーチの場合のみを説明します．

サーチのとき，現在サーチしているアドレスを，表示するかどうか聞いてきます．表示する場合は，[Y]を入力します(実行例4)．表示した場合，表示しない場合と比べ，サーチ時間が約2倍になります．

次にサーチデータ・バイト，1か2かを聞いてきます．

実行例1

```
 1.  SERCH OR DUMP

 2.  ﾀｰﾐﾅﾙ KEY ON (F.6 F.9 F.10)

 ( 1 OR 2 ) ? 2

 ﾀｰﾐﾅﾙ KEY OFF F.8 KEY !

 ( 1 OR 2 ) ? 1

***    ｻｰﾁ(1 OR 2 ﾊﾞｲﾄ) OR ﾀﾞﾝﾌﾟ    ***

BGIN ｱﾄﾞﾚｽ 16 ｼﾝ =? EA7C

END  ｱﾄﾞﾚｽ 16 ｼﾝ =? EABB

 ﾌﾟﾘﾝﾀｰ YES OR NO ? N

    SERCH OR DUMP ? D
```

実行例2

```
 BGIN=EA7C   END= EABB   DUMP

ｱﾄﾞﾚｽ  0 +1 +2 +3 +4 +5 +6 +7

EA7C  6D 6F 6E 0D 00 00 00 00 MON.....
EA84  00 00 00 00 00 00 00 00 ........
EA8C  3F 20 68 65 78 24 28 63 ?.HEX$(C
EA94  29 0D 00 00 00 00 00 00 ).......
EA9C  67 6F 20 74 6F 20 00 00 GO.TO...
EAA4  00 00 00 00 00 00 00 00 ........
EAAC  6C 69 73 74 20 00 00 00 LIST....
EAB4  00 00 00 00 00 00 00 00 ........
END
```

実行例 3

```
***   ｻｰﾁ(1 OR 2 ﾊﾞｲﾄ) OR ﾀﾞﾝﾌﾟ   ***

BGIN ｱﾄﾞﾚｽ 16 ｼﾝ =? 0

END  ｱﾄﾞﾚｽ 16 ｼﾝ =? 200

 ﾌﾟﾘﾝﾀｰ YES OR NO ? N

    SERCH OR DUMP ? S

ｻｰﾁ ｱﾄﾞﾚｽ ﾋｮｳｼﾞ YES OR NO ? N

ﾃﾞｰﾀ ﾊﾞｲﾄ 2(YES) OR 1(NO) ? N

     ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ YES OR NO? Y

     ｻｰﾁ ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ? @


BGIN=      0  HEX= 0000  SERCH
END =    512  HEX= 0200  DETA  40

  ｻｰﾁ ｱﾄﾞﾚｽ -3 -2 -1  0 +1 +2 +3

   26 001A   0C C3 A6 40 F3 0B C3
                ﾃ  ｦ  @  ﾓ     ﾃ
  182 00B6   5A 01 E6 40 CA 68 01
             Z     ﾆ  @  ﾊ  H
  204 00CC   C9 C9 DB 40 E6 08 EE
             ﾉ  ﾉ  ﾛ  @  ﾆ     ﾎ
  219 00DB   E6 FE D3 40 F6 01 D3
             ﾆ  "  ﾓ  @  ｮ     ﾓ
  223 00DF   F6 01 D3 40 32 67 EA
             ｮ     ﾓ  @  2  G  ﾊ
END
```

2バイトサーチなら **Y** を入力します．次にキャラクタ・コード入力か聞いてきます．キャラクタなら **Y** を入力します．キャラクタ入力の場合,1文字入力なので注意してください！

キャラクタ・コード以外の場合，サーチデータを16進で入力します(**実行例 5**)．サーチは，これで入力終わりです．あと終わるまで待ちます．

プログラム説明

プログラムは**リスト 1** に示します．

以下，プログラムの説明を行番号ごとに説明します．

実行例 4

```
***   ｻｰﾁ(1 OR 2 ﾊﾞｲﾄ) OR ﾀﾞﾝﾌﾟ   ***

BGIN ｱﾄﾞﾚｽ= ? 0

END  ｱﾄﾞﾚｽ= ? 5FFF

 ﾌﾟﾘﾝﾀｰ YES OR NO ? N

    SERCH OR DUMP ? S

ｻｰﾁ ｱﾄﾞﾚｽ ﾋｮｳｼﾞ YES OR NO ? Y

ﾃﾞｰﾀ ﾊﾞｲﾄ 2(YES) OR 1(NO) ? Y

     ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ YES OR NO ? Y

     ｻｰﾁ ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ=? F

NO.2 ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ YES OR NO ? Y

NO.2 ｻｰﾁ ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ=? O

 BGIN=0000  END= 5FFF  46  6F   5FFF

ｱﾄﾞﾚｽ -3 -2 -1  0 +1 +2 +3 +4 SERCH

1FEF  FB E1 C9 46 6F 75 6E 64 ﾛﾁ/FOUND
END
```

実行例 5

```
***   ｻｰﾁ(1 OR 2 ﾊﾞｲﾄ) OR ﾀﾞﾝﾌﾟ   ***

BGIN ｱﾄﾞﾚｽ= ? 0

END  ｱﾄﾞﾚｽ= ? FF

 ﾌﾟﾘﾝﾀｰ YES OR NO ? N

    SERCH OR DUMP ? S

ｻｰﾁ ｱﾄﾞﾚｽ ﾋｮｳｼﾞ YES OR NO ? N

ﾃﾞｰﾀ ﾊﾞｲﾄ 2(YES) OR 1(NO) ? N

     ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ YES OR NO ? N

     ｻｰﾁ ﾃﾞｰﾀ= ? 20


 BGIN=0000  END= 00FF  20

ｱﾄﾞﾚｽ -3 -2 -1  0 +1 +2 +3 +4 SERCH

00FB  A7 C1 F5 20 C5 18 C6 3E ｧﾁ1.ﾅ.ﾆ>
END
```

20行	最初のPOKE文は，キー表示のリバースをなくすものです．次の画面のクリア時に反リバースになります． もう1つPOKEは，ヌル・キャラクタを20Hにするためのものです．これは画面プリントの際に，00H(通常)になっていると文字の間が詰まって，うまくプリントされないことがあったため入れました．
30行	A～Nを整数変数に，R～Zを文字変数にします．
40行	KEY2の，?HEX$(C)+CHR$(13)は，画面に現在サーチしているアドレスを表示していない場合に使用するものです．STOP キーで中断させて，このキーを使用すると，今サーチしているアドレスを表示します．続行の場合は，CTRL RETURN です．
110行	PC-8001のターミナル・モードの f･6 f･9 f･10 をOKにするものです． f･6　コントロール・コード表示 f･9　画面のページ・プリント f･10　ライン・プリンタの改行 f･8　で，このモードOFF
200～220行	BGIN，ENDアドレスを入力します．
230～250行	プリンタを使用するかどうか？　使用するならK=1として，プリンタを40桁にします．
260～280行	サーチかダンプにするかチェック．CをBGINアドレスにします．
290行	ダンプならば(GOSUB670)BGIN，END表示．サーチなら310行へ．
300行	ダンプを横8バイトずつ表示．現在のアドレス+8を繰り返します．
310～320行	サーチ中のアドレスを表示するかどうかチェック．表示する場合A=1
330～370行	サーチデータ・バイトが2バイトならL=2，1バイトならL=1
380行	サーチの終わりかどうか調べます．終わりなら950行へ
400行	サーチデータが2バイトのときのチェック．あれば430行へ
410行	サーチデータが1バイトのときのチェック．あれば430行へ
420行	サーチデータがなければ，Cを+1して380行へ
430行	サーチのときの8バイトの表示へ．終わったら380行へ
440～600行	サーチデータ入力用サブルーチン
610～660行	サーチ中のアドレスを画面の右上に表示するためのサブルーチン
670～790行	サーチおよびダンプのBGIN，ENDのアドレス表示．またサーチなら，サーチデータの表示をします．
800～940行	ダンプの場合アドレス表示．8バイトずつのHEXおよびキャラクタ表示，ENDの判断をします．サーチの場合，サーチアドレス－3バイトから+4バイトまでのHEXおよびキャラクタの表示をします．
950行	END表示，プリンタを80桁にします．

リスト1　サーチ・ダンプ　プログラム・リスト

```
10 PRINT CHR$(12):CONSOLE 0,25,1,0
20 POKE &HEA5C,0:POKE &HEA5A,&H20
30 WIDTH40,25:DEFINT A-N:DEFSTR R-Z
40 KEY 1,"MON"+CHR$(13):KEY 2,"? HEX$(C)
"+CHR$(13)
50 PRINT " 1. SERCH OR DUMP":PRINT
60 PRINT " 2. ﾀｰﾐﾅﾙ KEY ON (F.6 F.9 F.10
)":PRINT
70 PRINT " ( 1 OR 2 ) ? ";
80 R=INKEY$:IF R="" THEN 80 ELSE PRINT R
90 IF R="1" OR R="2" THEN 100 ELSE 70
100 IF R="1" THEN 120
110 POKE &HEA58,1:PRINT:PRINT " ﾀｰﾐﾅﾙ KE
Y OFF F.8 KEY !":PRINT:GOTO 70
120 REM ******************************
130 REM ***                        ***
140 REM ***     SERCH OR DUMP      ***
150 REM ***      BY. H.ISHII       ***
160 REM ***       1979.11.25       ***
170 REM ***                        ***
180 REM ******************************
190 PRINT CHR$(12):CONSOLE 4
200 PRINT"***   ｻｰﾁ(1 OR 2 ﾊﾞｲﾄ) OR ﾀﾞﾝﾌ
ﾟ   ***":PRINT
210 INPUT "BGIN ｱﾄﾞﾚｽ= ";Y:B=VAL("&H"+Y)
:PRINT
220 INPUT "END  ｱﾄﾞﾚｽ= ";Y:E=VAL("&H"+Y)
:PRINT
230 IF E=<B THEN 200 ELSE PRINT" ﾌﾟﾘﾝﾀｰ
YES OR NO ? ";
240 S=INKEY$:IF S="" THEN 240 ELSE PRINT
 S
250 IF S="Y" THEN K=1:LPRINT CHR$(30)
260 PRINT :PRINT "    SERCH OR DUMP ? ";
:C=B
270 T=INKEY$:IF T="" THEN 270 ELSE PRINT
 T
280 IF K=1 THEN LPRINT"SERCH OR DUMP ? "
;T
290 IF T<>"S" THEN GOSUB 670 ELSE 310
300 GOSUB 800:C=C+8:GOTO 300
310 PRINT :PRINT "ｻｰﾁ ｱﾄﾞﾚｽ ﾋｮｳｼﾞ" YES OR
 NO ? ";
320 U=INKEY$:IF U="" THEN 320 ELSE PRINT
 U:IF U="Y" THEN A=1 ELSE A=0
330 PRINT :PRINT "ﾃﾞｰﾀ ﾊﾞｲﾄ 2(YES) OR 1(
NO) ? ";
340 V=INKEY$:IF V="" THEN 340 ELSE PRINT
 V:IF V="Y" THEN L=2 ELSE L=1
350 IF K=1 THEN LPRINT"ﾃﾞｰﾀ ﾊﾞｲﾄ 2(YES)
OR 1(NO) ? ";V
360 GOSUB 440:M=G:IF L=2 THEN L1=2:GOSUB
 440:N=G
370 H=1:GOSUB 670
380 IF C>E THEN 950
390 D=PEEK(C):IF A=1 THEN GOSUB 610
400 IF L=2 THEN IF D=M THEN C=C+1:D=PEEK
(C):IF D=N THEN C=C-1:GOTO 430
410 IF L=1 THEN IF D=M THEN 430
```

```
420 C=C+1:GOTO 380
430 GOSUB 800:C=C+4:GOTO 380
440 PRINT :IF L1=2 THEN PRINT "NO.2 ";EL
SE PRINT "      ";
450 PRINT "ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ YES OR NO ? ";
460 W=INKEY$:IF W="" THEN 460 ELSE PRINT
470 IF K=1 THEN IF L1=2 THEN LPRINT "NO.
2 ";ELSE LPRINT "      ";
480 IF K=1 THEN LPRINT"ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ YES
 OR NO ? ";W
490 IF W="Y" THEN 550
500 PRINT :IF L1=2 THEN PRINT "NO.2 ";EL
SE PRINT "      ";
510 INPUT "ｻｰﾁ ﾃﾞｰﾀ= ";Y:G=VAL("&H"+Y)
520 IF K=1 THEN IF L1=2 THEN LPRINT "NO.
2 ";ELSE LPRINT "      ";
530 IF K=1 THEN LPRINT"ｻｰﾁ        ﾃﾞｰﾀ="
;HEX$(G)
540 RETURN
550 PRINT :IF L1=2 THEN PRINT "NO.2 ";EL
SE PRINT "      ";
560 PRINT"ｻｰﾁ ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ=? ";
570 X=INKEY$:IF X="" THEN 570 ELSE PRINT
 X
580 IF K=1 THEN IF L1=2 THEN LPRINT "NO.
2 ";ELSE LPRINT "      ";
590 IF K=1 THEN LPRINT"ｻｰﾁ ｱｽｷｰｺｰﾄﾞ ﾃﾞｰﾀ
=? ";X
600 G=ASC(X):RETURN
610 Z=RIGHT$("000"+HEX$(C),4)
620 POKE&HF346,(ASC(MID$(Z,4,1)))
630 POKE&HF344,(ASC(MID$(Z,3,1)))
640 POKE&HF342,(ASC(MID$(Z,2,1)))
650 POKE&HF340,(ASC(MID$(Z,1,1)))
660 RETURN
670 PRINT CHR$(12):LOCATE 0,0
680 PRINT USING " BGIN=&  & ";RIGHT$("00
0"+HEX$(B),4);:PRINT USING " END= &  &
";RIGHT$("000"+HEX$(E),4);
690 IF K=1 THEN LPRINT :LPRINT USING " B
GIN=&  & ";RIGHT$("000"+HEX$(B),4);:LPRI
NT USING " END= &  &  ";RIGHT$("000"+HEX
$(E),4);
700 IF T="S" THEN 710 ELSE PRINT "DUMP";

710 IF K=1 THEN IF T="S" THEN 720 ELSE L
PRINT "DUMP";
720 IF T="S" THEN PRINT RIGHT$("0"+HEX$(
M),2);
730 IF K=1 THEN IF T="S" THEN LPRINT RIG
HT$("0"+HEX$(M),2);
740 IF L=2 THEN PRINT "  ";RIGHT$("0"+HE
X$(N),2) ELSE PRINT
750 IF K=1 THEN IF L=2 THEN LPRINT "  ";
RIGHT$("0"+HEX$(N),2) ELSE LPRINT
760 PRINT
770 IF T<>"S" THEN PRINT "ｱﾄﾞﾚｽ  0 +1 +2
 +3 +4 +5 +6 +7";:IF K=1 THEN LPRINT:LPR
INT "ｱﾄﾞﾚｽ  0 +1 +2 +3 +4 +5 +6 +7":LPRI
NT
780 IF T="S" THEN PRINT "ｱﾄﾞﾚｽ -3 -2 -1
 0 +1 +2 +3 +4 SERCH";:IF K=1 THEN LPRIN
T:LPRINT "ｱﾄﾞﾚｽ -3 -2 -1  0 +1 +2 +3 +4
SERCH":LPRINT
790 PRINT:LOCATE 0,4:RETURN
800 PRINT USING "&   &";RIGHT$("000"+HEX
$(C),4);
810 IF K=1 THEN LPRINT USING "&   &";RIG
HT$("000"+HEX$(C),4);
820 IF H=1 THEN C=C-3
830 FOR C1=0 TO 7:D=PEEK(C+C1)
840 IF H<>1 THEN IF C+C1>E THEN PRINT "
  ";:IF K=1 THEN LPRINT "   ";:GOTO 870
ELSE GOTO 870
850 PRINT USING " &&";RIGHT$("0"+HEX$(D)
,2);
860 IF K=1 THEN LPRINT USING " &&";RIGHT
$("0"+HEX$(D),2);
870 NEXT C1:PRINT " ";:IF K=1 THEN LPRIN
T " ";
880 FOR C2=0 TO 7:D=PEEK(C+C2)
890 IF D=<32 THEN PRINT ".";ELSE PRINT C
HR$(D);
900 IF K=1 THEN IF D=<32 THEN LPRINT " "
;ELSE LPRINT CHR$(D);
910 IF H<>1 THEN IF C+C2=E THEN GOSUB 93
0:GOTO 950
920 NEXT C2
930 PRINT :IF K=1 THEN LPRINT
940 RETURN
950 IF K=1 THEN LPRINT "END":LPRINT CHR$
(29)
960 PRINT "END":END
```

使用変数

プログラムで使用する変数は次のとおりです.

単純変数	A	サーチ中のアドレス表示用．yesのとき A=1
	B	BGINアドレス
	C	メイン・ルーチンでの現在のアドレス
	C1	HEX表示用 (FOR~NEXT用)
	C2	キャラクタ表示用 (FOR~NEXT用)
	D	PEEKしたときのCアドレスのデータ
	E	ENDアドレス
	G	サーチデータ入力用
	H	H=1のとき，サーチ
	K	プリンタ ONのとき，K=1
	L	サーチデータ・バイト．L=2(2バイト)
	L1	L1=2のとき no.2 入力用
	M	サーチデータ（1つ目用）
	N	サーチデータ（2つ目用）

ストリング変数	R	1.サーチ・ダンプ 2.ターミナル・キーON	
	S	プリンタ yes，no 入力用	Y のみ判別
	T	サーチ・ダンプ判別用	S のみ判別
	U	サーチ中のアドレス表示 yes，no	Y のみ判別
	V	サーチ・バイト yes，no用	Y のみ判別
	W	キャラクタ・コード yes，no	Y のみ判別
	X	キャラクタ・コード入力用	
	Y	16進入力用	
	Z	サーチ中のアドレス表示用	

モニタおよびサブルーチン

PC-8001のモニタ内のサブルーチンは次のようになっています.

アドレス	内　容
0F75H	キー入力サブルーチン
0257H	出力サブルーチン
5C2CH	モニタ・スタート・アドレス
5C3CH 5C66H	（5C2CHおよび5C3CHを，コールした場合には，コントロールBでN-BASICに戻れないので注意）
5C99H	Sコマンド
5D16H	Dコマンド
5D68H	Gコマンド
5D74H	Wコマンド
5DAEH	Lコマンド　および　LIコマンド
5DE6H	TMコマンド
5C5EH	0DH，0AH，3FHを出力，モニタへ
5E39H	HEXチェック・サブルーチン
5FADH	キー入力，出力サブルーチン（英小文字は英大文字に変える，03Hでモニタへ）
5FB9H	キー入力サブルーチン（英小文字は英大文字に変える．03Hでモニタへ出力しない）

●スタック

EA57H	00H以外で現在のカーソルのある位置より1つ下の行をクリアする．
EA58H	00Hノーマル，00H以外でターミナル・モードの f·6 f·9 f·10 使用可． 01Hを入れて f·8 でクリアできる．
EA5AH	ヌルキャラクタ・コード　普通　00H
EA5BH	アンダーライン，リバース他用，普通00H　白黒モードのときアトリビュート・エリアに入る
EA5CH	普通50H，これを00Hにして画面をクリアするとキーファンクションのリバースがなくる．
EA5DH	スクロール幅 00Hで1行，01Hで2行．
EA5EH	スクロール位置　01Hでトップ．
EA5FH	カーソルON OFF用　00H以外表示．
EA60H	キーファンクション表示ON OFF用．00H以外表示．
EA61H	グラフィックSW　ON OFF用．
EA62H	画面上下の幅20行で14H，25行で19H．
EA63H	カーソルの現在の縦の行．
EA64H	カーソルの現在の横の行．
EA65H	画面横幅のソフト切り替え用．80文字で50H，72文字で48H，40文字で28H，36文字で24Hが入っています． ハードの方は，ポート30Hに出力して，CRTCをセットする．
EA66H	ポート30Hセット用．行40のとき06Hが入っている．
ポート30H	リレーのON OFFに使用．画面（CRTC）40，80の切り替えに使用している．



PC−8001

また，ポート30Hに出力するものにより，次のように変わります．

OUTデータ	モータ	画面
0EH	ON	40
06H	OFF	40
0FH	ON	80
07H	OFF	80

EA66Hにも，データを入れてください．

●キーの内容の入っているアドレス

EA7CH──EA8BH	KEY1
EA8CH──EA9BH	KEY2
EA9CH──EAABH	KEY3
EAACH──EABBH	KEY4
EABCH──EACBH	KEY5
EACCH──EADBH	KEY6
EADCH──EAEBH	KEY7
EAECH──EAFBH	KEY8
EAFCH──EB0BH	KEY9
EB0CH──EB1BH	KEY10
EF36H──EF3BH	セーブ，ロードときのネームの入るアドレス．
EF3CH──EF41H	現在ロード中のテープのネームが入るアドレス．

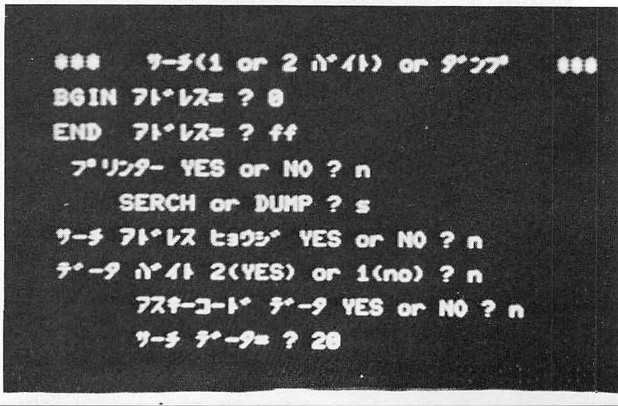
参考プログラム

1.プログラムを作ったときのブレーク・ルーチン

PC-8001の ESC キーが押されていたら，キャリーをセットするものです．

DB　09	IN	A, (9)	ポート
EE　7F	XOR	7FH	
C0	RET	NZ	
37	SCF		キャリーセット
C9	RET		

プログラム実行中の一例



ESC キー以外にしたい場合は，ポート番号およびXORの7FHを変えます．

2.キーI/Oチェック・プログラム

リスト2のプログラムを入れてください．このプログラムは，どのキーを押したら，何番のポートが，何々に変わるか調べるためのものです(**実行例6**)．

実行例6

```
PC-8001 KEY ｺｰﾄﾞ OR  KEY I/O ﾁｪｯｸ

F  F  F  F  F  F  F  F  F  7   HI
F  F  F  F  F  F  F  F  F  F   LO

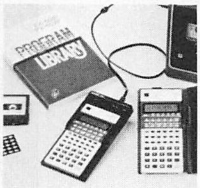
0  1  2  3  4  5  6  7  8  9
INPUT PORT ADDRESS
                    04:12:34
```

* * *

リスト2　キーI/Oチェック・プログラム

```
10 PRINT CHR$(12):POKE&HEA63,1
20 REM   *****************************
30 PRINT     "PC-8001 KEY ｺｰﾄﾞ OR  ";
40 PRINT          "KEY I/O ﾁｪｯｸ  "
50 REM   *          BY H.ISHII      *
60 REM   *          1979.12.11      *
70 REM   *****************************
80 WIDTH40:CONSOLE0,25,1,0
90 POKE&HEA63,3:POKE&HEA64,32
100 PRINT "HI"
110 POKE&HEA63,4:POKE&HEA64,32
120 PRINT "LO"
130 POKE&HEA63,6
140 PRINT "0  1  2  3  4  5  6  7  8  9"
150 PRINT "INPUT PORT ADDRESS"
160 POKE&HEA63,8:POKE&HEA64,20
170 PRINT TIME$:FOR A=0 TO 9
180 A$=RIGHT$("0"+HEX$(INP(A)),2)
190 POKE&HF468+6*A,(ASC(MID$(A$,2,1)))
200 POKE&HF3F0+6*A,(ASC(MID$(A$,1,1)))
210 NEXT A:GOTO 160
```

プログラム電卓のデータをPC-8001でリスト・アップ!



fx-502P DATA PRINTOUT PROGRAM

データ・プリントアウト・プログラム

井上　聡

最近，I/O誌には毎月『電卓コーナー』が載っていますが，そこに登場するのはたいていfx-502Pです．

この電卓は大変コスト-パフォーマンスは良いのですが，プリンタが使えないため不便に思っている人も多いようです．面白いプログラムを作っても，リストがないとI/O 誌に発表することもできません．

そこで，fx-502Pのプログラムやメモリ・データがKCS（カンサス・シティ・フォーマット）で保存されていることを利用して，ＰＣでそれを読み込んで，解析するプログラムを作ってみました．

「PC-8001で300ボーKCSが読み込める」

'79年にPC-8001の発売がアナウンスされたとき，当初のカタログではカセット・インターフェイス600/300ボーとなっていたのを覚えている方も多いでしょう．

事実，ポート30Hのビット４によって，これを切り換えることができるのです．これを利用して，ハードに何ら手を加えることなしに，ＫＣＳ規格のテープを読み込むことができます．

「fx-502Pのテープ・フォーマット」

メモリ・データをセーブした場合，図１のようになります．各メモリは８バイトで表わされ(図２)，ファイル・ネームはＦまたはＦＰの区別と，３桁の整数で図３のように表わされています．

プログラムの方は図４の型式ですが，それぞれコードが命令と対応しており，どれがどれと対応しているかはプログラムの行番号40，45と1000～4000を読めばわかるようになっています．

「プログラムの説明」

カセット・テープレコーダをＰＣに接続して，fxのテープを入れて再生してください．ＲＵＮするとプリンタへ出力するか，プログラムでＩＮＶを表示するかを聞いてくるので答えてください．なお，Ｐ５～Ｐ９に対し，ＩＮＶを付けると見にくくなるので，これは省いています．

図１　メモリ・データのセーブ・フォーマット

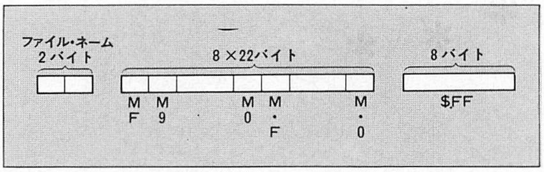


図２　各メモリのフォーマット

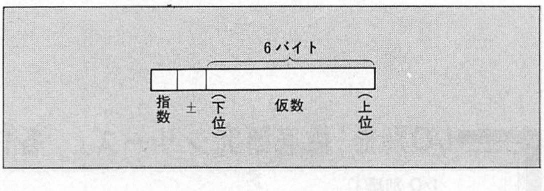


図３　ファイル・ネーム

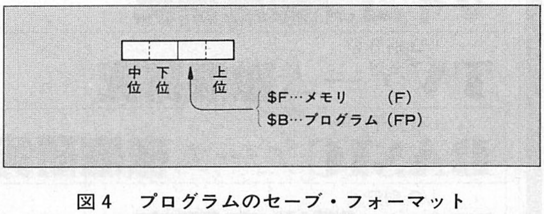
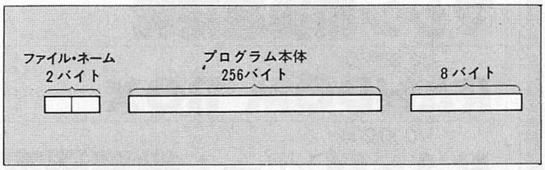


図４　プログラムのセーブ・フォーマット



後はタイミングを見計らって，何かキーを押すとテープを読み始めます．このタイミングは重要で，テープの頭の〝ピー〟という部分で押してください．うまくいけば数10秒後にテープの内容により，メモリ・データならＦが，プログラムならＦＰが表示され，ファイル・ネームも出ます．

これが，３桁の整数でなければ失敗なのでやり直してください．メモリの表示は１画面で収まりますが，プログラムの方は収まらないこともあるので，そのときは何かキーを押せばもう１度見ることができます．ESC キーを使って適当なところで止めるといいでしょう．

表1　プログラムの説明

行番号	内容
0～55	イニシャライズ
60～100	配列にテープから読み込む
1000～2000	メイン・プログラム
2000～5000	プログラム解析
5000～7000	メモリ・データ解析
8000～	命令コード・テーブル

表2　変数の説明

変数名	内容
A(257)	テープの内容
S(9), T(15, 5)	命令コード・テーブル
CR	改行時
IV	INV表示コード
IL	プリンタ出力
ID	INV表示するか

写真1　INVを表示しないときのリスト例

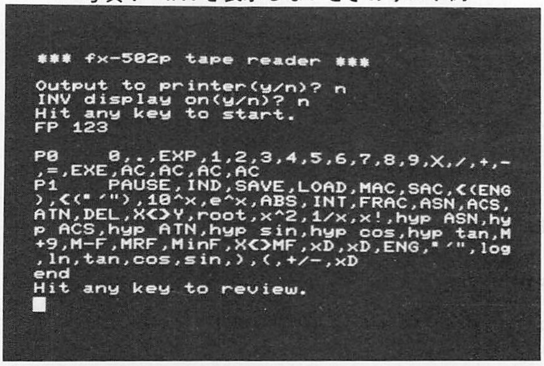


写真3　reviewを指示したときの例

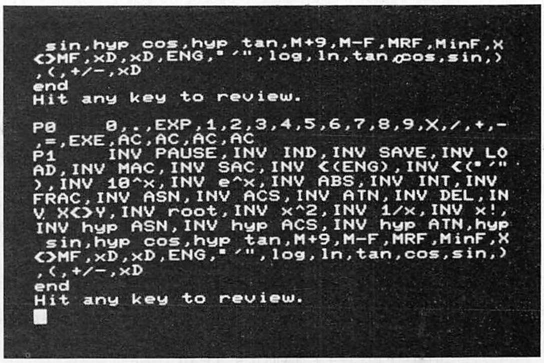


写真2　INVを表示したときのリスト例

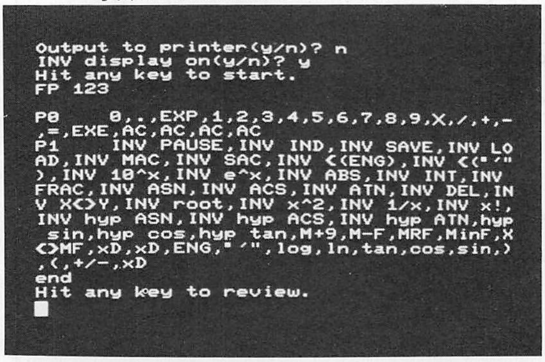
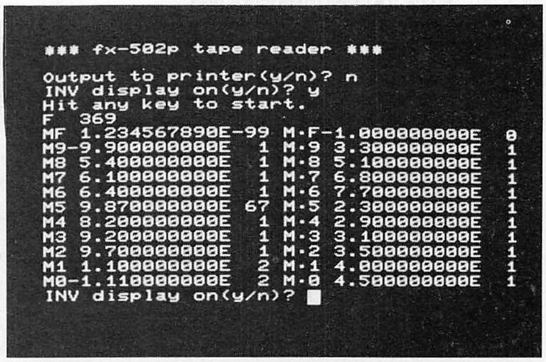


写真4　メモリの表示例



fx-502P用プログラムのプリント例

```
P0     0,.,EXP,1,2,3,4,5,6,7,8,9,x,/,+,-,=,EXE,AC,AC,AC,AC,
P1     INV PAUSE,INV IND,INV SAVE,INV LOAD,INV MAC,INV SAC,INV <(ENG),INV <(°'"),
INV 10^x,INV e^x,INV ABS,INV INT,INV FRAC,INV ASN,INV ACS,INV ATN,INV DEL,INV X<
>Y,INV root,INV x^2,INV 1/x,INV x!,INV hyp ASN,INV hyp ACS,INV hyp ATN,hyp sin,h
yp cos,hyp tan,M+9,M-F,MRF,MinF,X<>MF,xD,xD,ENG,°'",log,ln,tan,cos,sin,),(,+/-,x
D,
```

PC-8001版fx-502P用データ・プリントアウト・プログラム

```
5 CLEAR1000
10 CONSOLE0,25,0,1:WIDTH40,20:COLOR4:PRINT CHR$(12);:PRINT"*** fx-502p tape read
er ***":PRINT
20 DEFINTA-R:DEFSTRS-W:DIM A(257),S(9),T(15,5)
30 OUT33,255:OUT33,&H4E:OUT33,&H27:OUT48,16:OUT48,24
40 FOR J=0 TO 9:READ S(J):NEXTJ
45 FOR J=0 TO 15:FOR I=0 TO 5:READ T(J,I):NEXTI,J
50 INPUT "Output to printer(y/n)";W:IF W="y" THEN IL=1
55 GOTO 1000
60 FOR J=0 TO E
70 IF (INP(33)AND2)=0 THEN 70
80 A(J)=INP(32)
100 NEXT J:RETURN
999 REM *** KCS read ***
1000 INPUT "INV display on(y/n)";W:IF W="y" THEN ID=1
1005 PRINT "Hit any key to start."
1010 W=INPUT$(1):A=INP(32)
1020 E=257:GOSUB60
```



PC-8001版*fx-502P*用データ・プリントアウト・プログラム

```
1030 IF A(1)<240 THEN P=1:E=255:PRINT "FP"; ELSE E=175:PRINT "F ";
1050 PRINT STR$(A(1)MOD16)+RIGHT$("0"+HEX$(A(0)),2)
1080 IF P=1 THEN 2000 ELSE 5000
2000 REM *** program read ***
2010 FOR J=2 TO 257
2020 W="":IV=0:CR=0
2030 A=A(J):H=A¥16:L=AMOD16
2040 IF L<10 THEN 2500
2050 IF (H>5 AND H<13 AND A<>&HCA) OR H=14 OR A=&HFB THEN IV=1
2060 W=T(H,L-10):IF W="" THEN W="???"
2065 IF A=255 GOTO 4060
2070 GOTO 4000
2500 IF H=15 THEN 3000
2510 IF H=1 THEN IV=1
2520 IF H=0 THEN CR=1
2525 IF H=2 THEN CR=2
2530 IF H>9 THEN W=S(H-5)+"・" ELSE W=S(H)
2540 W=W+RIGHT$(STR$(L),1):GOTO 4000
3000 W=S(5+LMOD5):IF L¥5=1 THEN W=W+"・"
3010 W=W+"F":GOTO 4000
4000 IF ID=1 THEN IF IV=1 THEN W="INV "+W
4010 IF CR<>0 THEN PRINT CHR$(29);" ":W=SPACE$(CR-1)+W+SPACE$(7-CR*3):IF IL=1 TH
EN LPRINT " "
4020 PRINT W;:IF IL=1 THEN LPRINT W;
4040 IF CR=0 THEN PRINT ",";:IF IL=1 THEN LPRINT ",";
4050 NEXTJ
4060 PRINT CHR$(29);" ":PRINT "end":PRINT "Hit any key to review.":W=INPUT$(1):G
OTO 2000
5000 REM *** memory read ***
5010 FORI1=0TO10:W="M"
5020 IFI1=0THENW=W+"F"ELSEW=W+RIGHT$(STR$(10-I1),1)
5030 I=I1*8+2:GOSUB6000
5040 W=W+" M・"
5060 IFI1=0THENW=W+"F"ELSEW=W+RIGHT$(STR$(10-I1),1)
5070 I=I1*8+90:GOSUB6000:PRINTW:IFIL=1THENLPRINT W
5080 NEXTI1:GOTO 1000
6000 IF (A(I+1)AND8)=8 THEN W=W+"-" ELSE W=W+" "
6010 W=W+RIGHT$(STR$(A(I+7)),1)+"."
6020 FOR J=6 TO 2 STEP -1
6030 W=W+RIGHT$("0"+HEX$(A(I+J)),2)
6040 NEXT J
6050 W=LEFT$(W,LEN(W)-1)+"E"
6060 IF (A(I+1)AND1)=0 THEN IFA(I)<>0 THEN W=W+"-":W=W+RIGHT$(" "+STR$(100-VAL(H
EX$(A(I)))),2):RETURN
6070 W=W+RIGHT$("  "+STR$(VAL(HEX$(A(I)))),3):RETURN
8000 DATA P,RND,LBL,GOTO,GSBP
8010 DATA X<>M,Min,MR,M-,M+
8020 DATA 0,1,2,3,.,EXP
8030 DATA 4,5,6,7,8,9
8040 DATA HLT,,,,,
8050 DATA xD,,ENG,°'",log,ln
8060 DATA +/-,(,),sin,cos,tan
8070 DATA x,/,+,-,=,EXE
8080 DATA ,DSZ,x=0,x=F,RND#,(EXP)
8090 DATA ISZ,x>=0,x>=F,mean,sigma n,sigma n-1
8100 DATA PAUSE,IND,SAVE,LOAD,MAC,SAC
8110 DATA DEL,,<(ENG),<(°'"),10^x,e^x
8120 DATA ABS,INT,FRAC,ASN,ACS,ATN
8130 DATA x^y,x^1/y,R>P,P>R,%,(EXE)
8140 DATA GSB0,X<>Y,root,x^2,1/x,x!
8150 DATA MODE4,MODE5,MODE6,hyp sin,hyp cos,hyp tan
8160 DATA5,,,hyp ASN,hyp ACS,hyp ATN
8170 DATA AC,NOP,,,,end
10000 REM ****************************
10005 REM ***                      ***
10010 REM *** fx-502p disassembler ***
10015 REM ***                      ***
10020 REM *** By Time's lonely one ***
10025 REM ***                      ***
10030 REM ***         80/10/25     ***
10035 REM ***                      ***
10045 REM ****************************
```



RANDOM BOX

PC-8001 BASICのテキスト番号を"0"にする!

津のオケラ

I/O'80年10月号にDE JA3RZWさんのMZ-80用のプログラムを参考にしてPC-8001用のプログラムを作ってみました.

打ち込んだプログラムの後にこのプログラムを入れ, **RUN 10000**を実行してください. 修正されたプログラムが出力されます. 0番になったことを確かめた後**DELETE 1000—10040**を実行して, このプログラムを消去してください. **REM**文以外には作わないでください.

PC-8001ではBASICは次のように入っています.

```
8020  00
8021  45  80  次のライン・ナンバーのアドレス
8023  0A  00  ライン・ナンバー
  ⋮    ⋮
8045  A2  80
8047  14  00
  ⋮    ⋮
```

後は各自で考えてください. 別にそう難しくありません.

注) 16Kの人は10010行のI=&H8021をI=&HC021に変えてください.

```
10000 PRINT CHR$(12);"REM ｲｶﾞｲ ﾊ ﾂｶｴﾏｾﾝ"
10010 INPUT "ﾅﾝｷﾞｮｳﾒﾏﾃﾞ ｼﾏｽｶ ";A:I=&H8021
10020 POKEI+2,0:POKEI+3,0
10030 I=PEEK(I)+PEEK(I+1)*256
10040 IFPEEK(I+2)+PEEK(I+3)*256 >A THEN LIST:END ELSE GOTO 10020
```

■参考文献

1) 昭和一桁："SAVEミス救済法", I/O, '80年1月号
2) 清水の太郎長："スーパーコマンダー", I/O, '80年8月号
3) DE JA3RZW："MZ-80 BASICのテキスト番号を「0」にする", I/O, '80年10月号

PC-FAST

■渡辺 卓也

I/O '80年９月号のＭＺ用ＦＡＳＴをＰＣに移植しました．変更はほとんどしてません．文法はまったく同じです．

変更した点

ＭＺとＰＣではキャラクタ・コードやコントロール・コードが違うので，アセンブル・リストを参考にしてＰＣ用に変えました．ＭＺのモニタのルーチンをコールしているところは，ＰＣのモニタに同じものがあるときはそれを使い，ないときはサブルーチンを作りました．

番地の変更はメモリ・マップを見てください．8098Ｈ～80ＦＦＨは空いているので，予約語を増すときに使用できます．

なお，I/O '80年11月号のＲＡＮＤＯＭ ＢＯＸにあった変更はしてません．

PC-FASTの走らせ方

まず，ダンプ・リストどおりにキーインしてください．キーインが済んだら，

```
*W8020, 8B40 ⤶   (⤶はRETURNを押す)
```

として，テープにセーブしてください．

次に，

```
*G8100 ⤶
```

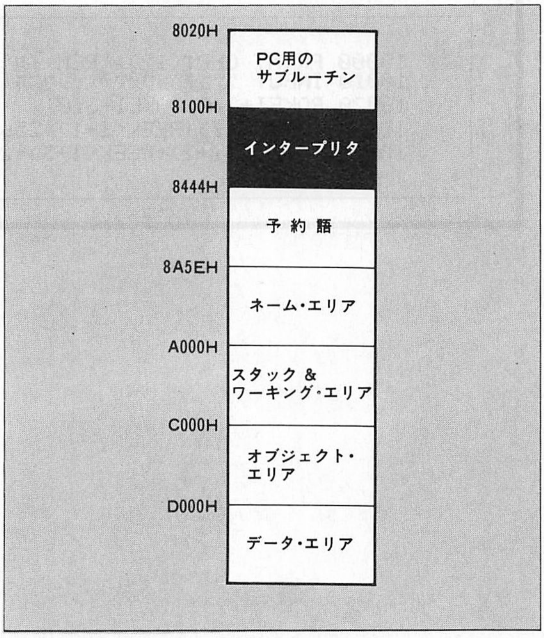
とすると，スタートします．ホット・スタートは812ＢＨからです．モニタに戻るには，CTRL Bです．

スタートすると『#』が出るので，プログラムをキーインしてください．キーインした文字は小文字→大文字の変換をしているので，SHIFTを押して大文字を打つ必要はありません．

プログラムの入力の方法

いま，『２＋３と４＊５を計算する』プログラムを考えます．このプログラムの名前をＡＢＣとしましょう．これは次のように入力します．

PC-FAST　メモリ・マップ




```
#〔⤶ ABC⤶ 2⤶ 3⤶ +⤶ 4⤶ 5⤶ *⤶ 〕⤶
```

途中で入力を止めるときは，％⤶と入力してください．

プログラムの実行

プログラムＡＢＣを実行するには，

```
#ABC⤶
```

とします．次に，

```
#STK⤶
```

とすると，いま実行したＡＢＣの結果がスタックにあることがわかります．また，このプログラムを他のプログラムから呼ぶこともできます．

たとえば，『(２＋３)＊(４＊５)』を計算するプログラムの名前をＤＥＦとすると，

```
#〔DEF  ABC  ＊ 〕
```

となります．ここで，

```
#DEF↲
```

とすると，まず，プログラムABCが実行され，スタックに00005と00020が積まれ，次の『＊』で結果の00100がスタックに積まれます．

このように，ほとんどの場合，他のプログラムとのパラメータのやりとりはスタックによって行なわれます．しかし，変数を使うことも可能です．

変数は使う前に宣言をしなければなりません．DATAという名前の変数の宣言は次のようにします．

```
#VAR↲  NAME?(下線部はコンピュータから)
#DATA↲
```

この変数に値を代入したいときは，

```
#DATA↲
#!↲
```

とします．これでスタックのTOPの値が変数『DATA』に代入されます．逆に変数の値をスタックにPUSHするには，

```
#DATA↲
#@↲
```

とします．また，配列も使えます．

```
#ARY↲  NAME?
#DIM↲  SIZE?
#5↲
```

この配列の名は『DIM』で，大きさは10バイトです．この配列の２番にスタックTOPの値を代入するプログラムは次のようになります．

```
#〔↲LET↲100↲2↲DIM↲!↲〕↲
```

２番を表示するプログラムは，

```
#〔↲PR↲2↲DIM↲@↲.↲〕↲
```

です．文字変数，文字配列も同じようになります．ただし，代入するときは『C!』をPUSHするときは『C@』を使います．宣言はそれぞれ『CVAR』，『CARY』を使います．

プログラムの中断は**ESC**を押してください．

簡単な応用例

I/O'80年12月号MZ用LISPの記事を参考に，『サルのバナナ取り』と『ハノイの塔』をFASTでプログラムしました（**リスト1,2**）．

最後に

９月号の記事を読んだときからPCに移植したいと思っていましたが，思うだけで，実際には何もしませんでした．ところが12月号で，クレージーバルーンがMZ用に移植された記事を読み，作った本人でさえ，忘れたような細かいところまで，プログラムを解読してあるのをみて驚きました．

そこで，負けてはいられないと思い，FASTのアセンブル・リストを自分がZ80になったつもりでたどってみました．そのとき，意味のわかりやすいラベルが役に立ちました．

また，ゲームなどと違って，ビデオRAMの番地に注意する必要がなかったので，思っていたより容易にできました．

PCのモニタの便利なルーチンは参考文献2)，3)，6)によりました．**ESC**を押すと中断するところのサブルーチンは参考文献3)によります．また，ハノイの塔は参考文献1)のPALL用のプログラムをFAST用にしました．

最後に，わかりやすいラベル名の付いたアセンブル・リ

写真１　ダイレクト・モードによる演算とスタック操作

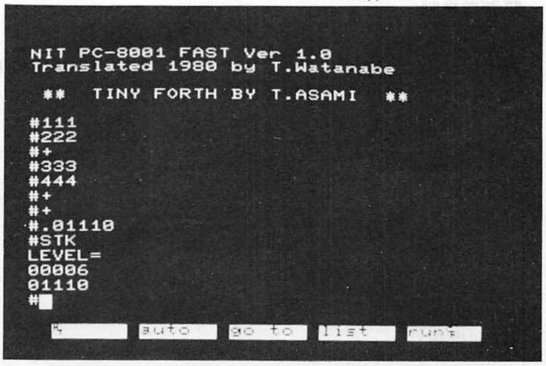


写真２　プログラム・モードの例

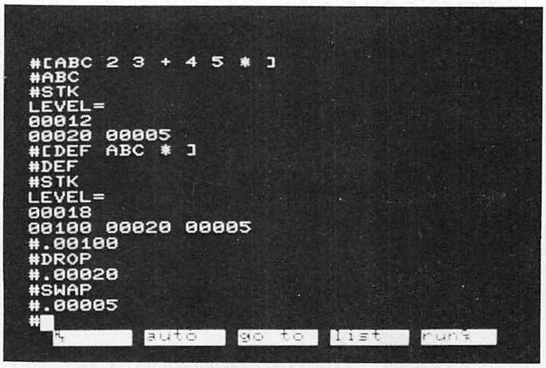
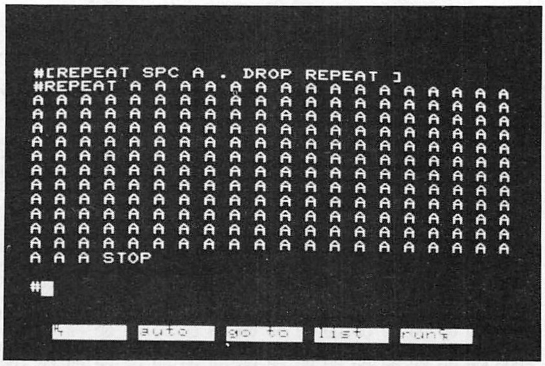


写真３　ワード『REPEAT』の実行例



ストを作成してくださった浅見氏に深く感謝いたします．

◘参考文献

1)藤原博文：〝ハノイの塔″，I/O，1979年12月号

2)桝田鷹男：〝BASICによる逆アセンブラ″，I/O，1980年2月号

3)石井晴正：〝サーチダンププログラム″，I/O，1980年，3月号

4)片桐　明：〝FORTH″，I/O，1980年6月号

5)浅見俊幸：〝FAST″，I/O，1980年9月号

6)長谷正博：〝PC-8001モニタサブルーチンの利用法″，I/O，1980年11月号

7)浅見俊幸・古川徹：〝LISP″，I/O，1980年12月号

(注）ＮＩＴとは名古屋工業大学の略です．

リスト1　サルのバナナ取り

a）プログラム・リスト

```
#VAR NAME?
MONKEY
#VAR NAME?
TABLE
#VAR NAME?
BANANA
#[S SPC . DROP ]
#[Q WHERE S IS S THE S ]
#[IN Q BANANA? S ? LF Q MONKEY? S ? LF Q
TABLE? S ? LF ]
#[NEXT MONKEY @ 1 ; = IF 2 ROOM2 ROOM1 E
LSE 1 ROOM1 ROOM2 THEN LF THE S 'MONKEY
S MOVES S FROM S S TO S S MONKEY ! ]
#[BRING NEXT WITH S THE S 'TABLE S MONKE
Y @ TABLE ! ]
#[SUB LF IN TABLE ! MONKEY ! BANANA ! ]
#[GAME SUB TABLE @ MONKEY @ = F,IF NEXT
THEN BANANA @ = F.IF BRING THEN
LF THE S 'MONKEY S ATE S THE S 'BANANA
S ]
```

b）実行例

```
#GAME ⤶
WHERE IS THE BANANA? 1 ⤶
WHERE IS THE MONKEY? 2 ⤶
WHERE IS THE TABLE? 1 ⤶
```

リスト2　ハノイの塔

a）プログラム・リスト

```
100 '  **  TINY FORTH BY T.ASAMI  **
110 '  #VAR NAME?
120 '  N
130 '  #CVAR NAME?
140 '  FROM
150 '  #CVAR NAME?
160 '  BY
170 '  #CVAR NAME?
180 '  TO
190 '  #[S SPC . DROP ]
200 '  #[TRNS LF MOVE S DISK S N @ S 'FROM S FR
210 '  OM C@ S 'TO S TO C@ S ]
220 '  #[MAIN TO C! BY C!iFROM C! N ! N @ 0 ; =
230 '  IF ELSE N @ FROM C@ BY C@ TO C@ N @ 1 -
240 '  FROM C@ TO C@ BY C@ MAIN TO C! BY C! FR
250 '  OM C! N ! TRANS N @ 1 - BY C@ FROM C@ TO
260 '  C@ MAIN THEN ]
270 '  #[HANOI TOWER-A TOWER-B TOWER-C MAIN ]
```

b）実行例

```
#3
#HANOI
MOVE DISK 00001 FROM TOWER-A TO TOWER-C
MOVE DISK 00002 FROM TOWER-A TO TOWER-B
MOVE DISK 00001 FROM TOWER-C TO TOWER-B
MOVE DISK 00003 FROM TOWER-A TO TOWER-C
MOVE DISK 00001 FROM TOWER-B TO TOWER-A
MOVE DISK 00002 FROM TOWER-B TO TOWER-C
MOVE DISK 00001 FROM TOWER-A TO TOWER-C
```

PC-FASTダンプ・リスト

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8000 E6 F1 E9 F1 C9 00 C9 00 7F 23 A9 23 C9 00 C9 00 :843
8010 C9 00 C9 00 C9 00 C9 00 C9 00 C9 00 C9 00 C9 00 :648
8020 00 00 C3 93 80 00 DB 09 EE 7F C9 C9 1A CD A6 40 :786
8030 13 FE 0D 20 F7 1B CD CA 5F C9 AF 21 FF A1 77 21 :817
8040 FF 9F 77 21 7F A0 77 DD 21 FF A1 31 FF EB 21 57 :8FD
8050 80 CD ED 52 C3 04 81 0C 4E 49 54 20 50 43 2D 38 :5E3
8060 30 30 31 20 46 41 53 54 20 56 65 72 20 31 2E 30 :3DB
8070 0D 1F 54 72 61 6E 73 6C 61 74 65 64 20 31 39 38 :500
8080 30 20 62 79 20 54 2E 57 61 74 61 6E 61 62 65 0D :4FD
8090 1F 1F 00 3E 20 C3 A6 40 00 00 00 00 00 00 00 00 :245
80A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
80B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
80C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
80D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
80E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
80F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8100 C3 3A 80 00 11 2B 00 ED 53 10 A1 11 40 8B ED 53 :5C6
8110 12 A1 11 FF BF ED 53 0C A1 11 FF CF ED 53 0E A1 :83D
8120 3E 1C CD A6 40 11 8F 81 CD 2C 80 CD CA 5F 3E 23 :6FE
8130 CD A6 40 CD 2B 80 11 00 A0 CD AD 81 CD DF 81 FE :902
8140 00 20 05 CD 24 82 18 E3 2A 00 A1 7E FE F1 30 05 :600
8150 CD 84 83 18 D6 20 05 CD AF 83 18 CF FE F2 20 05 :7E2
8160 CD 86 83 18 C6 FE F3 20 05 CD 9B 83 18 BD FE F4 :97C
8170 20 05 CD 86 83 18 B4 FE F5 20 05 CD 8B 83 18 AB :77D
8180 11 88 81 CD 2C 80 18 A3 20 45 52 52 20 31 0D 2A :4DF
8190 2A 20 20 54 49 4E 59 20 46 4F 52 54 48 20 42 59 :40C
81A0 20 54 2E 41 53 41 4D 49 20 20 2A 2A 0D CD 75 0F :3FF
81B0 CD C1 5F FE 0D 20 04 3E 0D 12 C9 FE 02 CA 66 5C :6CE
81C0 FE 1C 20 06 CD A6 40 13 18 E3 FE 1D 20 06 CD A6 :6B5
81D0 40 1B 18 D9 FE 20 38 D5 CD A6 40 12 13 18 CE C5 :6FA
81E0 D5 E5 ED 4B 10 A1 21 5E 8A 11 00 A0 79 FE 00 20 :6F4
81F0 0D 78 FE 00 20 08 ED 43 00 A1 E1 D1 C1 C9 1A BE :790

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8200 20 08 FE 0D 28 0C 23 13 18 F4 7E FE 0D 28 0F 23 :48C
8210 18 F8 23 4E 23 46 ED 43 00 A1 E1 D1 C1 C9 23 23 :73D
8220 23 0B 18 C5 3E 0D DD 23 DD 77 00 1B 1A FE 21 D8 :5D6
8230 18 F4 C5 E5 21 00 00 06 00 DD 7E 00 DD 2B FE 0D :64B
8240 20 06 22 00 A1 E1 C1 C9 FE 30 38 22 FE 3A 30 04 :648
8250 D6 30 18 0A FE 41 38 16 FE 47 30 12 D6 37 4F 29 :5C1
8260 38 0C 29 38 09 29 38 06 29 38 03 09 30 CB 11 77 :305
8270 82 CD 2C 80 C3 2B 81 20 45 52 52 20 32 0D C5 E5 :67C
8280 3E 0D DD 23 DD 77 00 21 00 A1 CD 9A 82 CD 9A 82 :733
8290 23 CD 9A 82 CD 9A 82 E1 C1 C9 AF ED 67 FE 0A 30 :99B
82A0 08 C6 30 DD 23 DD 77 00 C9 C6 37 18 F6 C5 2A 00 :715
82B0 A1 11 00 A0 01 10 27 CD D9 82 01 E8 03 CD D9 82 :6C6
82C0 01 64 00 CD D9 82 01 0A 00 CD D9 82 3E 30 85 12 :5C5
82D0 13 3E 0D 12 CD 24 82 C1 C9 AF 3E 30 ED 42 38 03 :5F4
82E0 3C 18 F9 09 12 13 C9 C5 16 00 21 00 00 DD 7E 00 :49B
82F0 FE 0D 28 0A FE 30 38 2A FE 3A 38 09 18 24 DD 2B :58A

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8300 22 00 A1 C1 C9 D6 30 29 38 18 22 00 A1 29 38 12 :502
8310 29 38 0F ED 4B 00 A1 5F 09 38 07 19 38 04 DD 2B :44D
8320 18 CB 11 2B 83 CD 2C 80 C3 2B 81 20 45 52 52 20 :5B3
8330 33 0D C5 D5 E5 DD 7E 00 FE 2D 28 07 CD E7 82 E1 :88B
8340 D1 C1 C9 DD 2B CD E7 82 AF ED 4B 00 A1 21 FF FF :A40
8350 ED 42 23 22 00 A1 E1 D1 C1 C9 C5 D5 E5 ED 4B 00 :906
8360 A1 78 A7 FA 6D 83 CD AD 82 E1 D1 C1 C9 AF 21 FF :AB1
8370 FF ED 42 23 22 00 A1 CD AD 82 DD 23 DD 36 00 2D :750
8380 E1 D1 C1 C9 23 E9 23 22 02 A1 C9 CD 32 83 ED 5B :8C3
8390 00 A1 EB 29 29 29 29 EB 19 18 EB CD 32 83 ED 5B :701
83A0 00 A1 EB 29 EB 19 18 DE 0A 6F 03 0A 67 7E C9 FD :6E0
83B0 21 FF AA 22 0A A1 ED 4B 0A A1 03 CD 26 80 28 0C :624
83C0 CD A8 83 FE F1 30 14 CD 84 83 18 EE 11 D5 83 CD :93B
83D0 2C 80 C3 2B 81 20 53 54 4F 50 0D 20 0C FD 23 FD :5D7
83E0 71 00 FD 23 FD 70 00 18 CA FE F2 20 05 CD 86 83 :7CB
83F0 18 C8 FE F3 20 05 CD 9B 83 18 BF FE F4 20 05 CD :89C

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8400 86 83 18 B6 FE F5 20 05 CD 8B 83 18 AD FE F6 20 :8A3
8410 13 23 11 00 A0 7E 12 FE 0D 28 04 23 13 18 F6 CD :4BF
8420 24 82 18 96 FE FF C2 6E 81 FD 22 00 A1 AF 2A 00 :79B
8430 A1 11 FF AA ED 52 C8 FD 46 00 FD 2B FD 4E 00 FD :915
8440 2B C3 BA 83 01 ED 5B 12 A1 ED 53 02 A1 ED 5B 0C :75E
8450 A1 ED 53 04 A1 ED 5B 0E A1 ED 53 06 A1 ED 5B 10 :7BC
8460 A1 ED 53 08 A1 13 ED 53 10 A1 ED 5B 12 A1 13 CD :769
8470 AD 81 EB ED 4B 0C A1 03 23 71 23 70 22 12 A1 60 :65D
8480 69 36 F1 CD 20 80 11 00 A0 CD AD 81 3A 00 A0 FE :781
8490 5D 20 0B 11 5D 8A 23 73 23 72 22 0C A1 C9 FE 27 :568
84A0 20 1F 11 01 A0 ED 4B 0E A1 03 23 71 23 70 3E F6 :536
```

PC-FASTダンプ・リスト

```
84B0 02 03 1A 02 FE 0D 28 03 13 18 F6 ED 43 0E A1 18 :46F
84C0 C2 FE 25 20 31 ED 5B 02 A1 ED 53 12 A1 ED 5B 04 :760
84D0 A1 ED 53 0C A1 ED 5B 06 A1 ED 53 0E A1 ED 5B 08 :7BC
84E0 A1 ED 53 10 A1 CD CA 5F 11 EF 84 CD 2C 80 C9 46 :894
84F0 4F 52 47 45 54 0D CD DF 81 FE 00 20 05 11 00 A0 :58F

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8500 18 A3 ED 4B 00 A1 23 71 23 70 C3 83 84 02 DD 22 :686
8510 00 A1 ED 5B 00 A1 1A FE 20 30 01 C9 CD A6 40 1B :68A
8520 18 F4 03 CD 20 80 C9 04 CD CA 5F C9 05 CD 77 85 :7D6
8530 3E F2 12 13 13 ED 53 0E A1 C9 06 CD 77 85 ED 53 :72F
8540 0A A1 11 68 85 CD 2C 80 11 00 A0 CD AD 81 CD 24 :6BF
8550 82 CD E7 82 ED 5B 0A A1 3E F3 12 2A 00 A1 29 38 :71A
8560 0F 19 38 0C 22 0E A1 C9 20 53 49 5A 45 3F 20 0D :3CD
8570 20 45 52 52 20 34 0D 11 9A 85 CD 2C 80 ED 5B 12 :56D
8580 A1 13 CD AD 81 EB ED 5B 0E A1 13 23 73 23 72 22 :6F1
8590 12 A1 2A 10 A1 23 22 10 A1 C9 20 4E 41 4D 45 3F :4CD
85A0 20 0D 07 AF DD 22 00 A1 2A 00 A1 11 FF A1 ED 52 :63E
85B0 22 00 A1 CD CA 5F 11 F2 85 CD 2C 80 CD AD 82 DD :893
85C0 7E 00 DD 2B FE 0D 28 05 CD A6 40 18 F2 CD CA 5F :771
85D0 DD 22 00 A1 ED 5B 00 A1 CD 26 80 C8 21 FF A1 ED :872
85E0 52 C8 1A 1B FE 20 38 05 CD A6 40 18 EB CD 20 80 :6CD
85F0 18 E6 4C 45 56 45 4C 3D 0D 08 CD CA 5F 01 5E 8A :5A7

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8600 CD 26 80 C8 2A 12 A1 ED 42 D8 0A 03 FE 0D 28 05 :664
8610 CD A6 40 18 EB CD A8 83 03 FE F1 30 08 11 66 86 :7D5
8620 CD 2C 80 18 DB 20 08 11 6E 86 CD 2C 80 18 D1 FE :6F9
8630 F2 20 08 11 76 86 CD 2C 80 18 C5 FE F3 20 08 11 :6A7
8640 7E 86 CD 2C 80 18 B9 FE F4 20 08 11 86 86 CD 2C :77E
8650 80 18 AD FE F5 20 08 11 8F 86 CD 2C 80 18 A1 11 :6C9
8660 88 81 CD 2C 80 C9 20 28 57 29 20 20 20 0D 20 28 :4C8
8670 50 29 20 20 20 0D 20 28 56 29 20 20 20 0D 20 28 :262
8680 41 29 20 20 20 0D 20 28 43 56 29 20 20 20 0D 20 :26E
8690 28 43 41 29 20 20 20 0D 09 CD 32 82 CD AD 82 C9 :591
86A0 0A CD 32 82 CD 5A 83 C9 0B CD E7 82 CD 7E 82 C9 :8D5
86B0 0C CD 32 83 CD 7E 82 C9 0D D5 11 00 A0 CD AD 81 :7B2
86C0 CD 24 82 D1 C9 0E 2A 02 A1 5E 23 56 ED 53 00 A1 :6A0
86D0 CD 5A 83 C9 0F 2A 02 A1 22 0A A1 CD 32 83 ED 5B :6E6
86E0 00 A1 2A 0A A1 73 23 72 C9 10 DD 21 FF A1 C9 11 :6CF
86F0 C5 DD 22 00 A1 ED 4B 00 A1 11 00 A0 0A 12 FE 20 :629

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8700 38 04 0B 13 18 F6 CD 24 82 C1 C9 12 C5 11 00 A0 :5ED
8710 CD 31 87 ED 53 02 A1 11 80 A0 CD 31 87.ED 53 04 :762
8720 A1 ED 5B 02 A1 CD 24 82 ED 5B 04 A1 CD 24 82 C1 :820
8730 C9 AF DD 22 00 A1 ED 4B 00 A1 21 FF A1 ED 42 28 :809
8740 0C DD 7E 00 12 DD 2B FE 0D C8 13 18 E4 11 56 87 :651
8750 CD 2C 80 C3 2B 81 20 45 52 52 20 36 0D 13 DD 22 :566
8760 00 A1 ED 5B 00 A1 AF 21 FF A1 ED 52 28 06 1A 1B :69C
8770 FE 0D 20 F2 ED 53 00 A1 DD 2A 00 A1 C9 CD 32 83 :7F1
8780 ED 5B 00 A1 CD 32 83 2A 00 A1 C9 14 CD 7D 87 19 :6FD
8790 22 00 A1 CD 5A 83 C9 15 CD 7D 87 AF ED 52 EA A8 :89C
87A0 87 22 00 A1 CD 5A 83 C9 11 B1 87 CD 2C 80 C3 2B :76D
87B0 81 20 45 52 52 20 38 0D 16 C5 CD 32 83 ED 4B 00 :584
87C0 A1 CD 32 83 21 01 A1 11 00 00 CD E0 87 CD E0 87 :75F
87D0 2B CD E0 87 CD E0 87 ED 53 00 A1 CD 5A 83 C1 C9 :9A8
87E0 EB 29 38 1A 29 38 17 29 38 14 29 38 11 AF EB ED :54C
87F0 6F EB FE 00 20 02 EB C9 09 38 03 3D 18 F4 11 07 :5D3
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8800 88 CD 2C 80 C3 2B 81 20 45 52 52 20 39 0D 17 C5 :5BB
8810 01 00 00 CD 7D 87 ED 53 00 A1 7A FE 00 20 15 7B :5DB
8820 FE 00 20 10 11 2D 88 CD 2C 80 C3 2B 81 20 45 52 :593
8830 52 20 37 0D 3E 0C CD 8E 88 38 0C AF ED 52 38 06 :553
8840 3E 10 80 47 18 F6 19 ED 5B 00 A1 3E 08 CD 8E 88 :64E
8850 38 09 AF ED 52 38 03 04 18 F9 19 ED 5B 00 A1 3E :5BF
8860 04 CD 8E 88 38 0C AF ED 52 38 06 3E 10 81 4F 18 :58D
8870 F6 19 ED 5B 00 A1 AF ED 52 38 03 0C 18 F9 19 22 :679
8880 00 A1 CD 5A 83 ED 43 00 A1 CD 5A 83 C1 C9 EB 29 :864
8890 38 03 3D 20 FA EB C9 DD 23 3E 0D DD 77 00 DD 23 :6E5
88A0 3E 31 DD 77 00 C9 DD 23 3E 0D DD 77 00 DD 23 3E :669
88B0 30 DD 77 00 C9 18 C5 11 00 A0 CD 31 87 ED 53 02 :6A2
88C0 A1 11 80 A0 CD 31 87 2A 02 A1 1A BE C2 DD 88 FE :821
88D0 00 CA D8 88 1B 2B 18 F2 CD 97 88 C1 C9 CD A6 88 :8EB
88E0 C1 C9 19 CD 7D 87 AF EB ED 52 30 04 CD 97 88 C9 :936
88F0 CD A6 88 C9 1A CD 7D 87 AF ED 52 30 04 CD 97 88 :8BD

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8900 C9 CD A6 88 C9 1B CD 32 83 ED 5B 00 A1 AF BA C0 :93C
8910 BB C0 03 CD A8 83 FE 1C C8 FE 1D C8 FE FF 20 F2 :A4A
8920 11 29 89 CD 2C 80 C3 2B 81 20 45 52 52 20 35 0D :516
8930 1C 03 CD A8 83 FE 1D C8 FE FF 20 F5 C3 20 89 1D :895
8940 C9 1E CD 32 83 ED 5B 00 A1 AF BA 20 C5 BB 20 C2 :83D
8950 C9 1F CD 32 83 ED 5B 00 A1 AF BA C0 BB C0 0B 0B :80D
8960 0B CD A8 83 FE 20 C8 FE FF 20 F4 C3 20 89 20 C9 :94F
8970 21 CD 7D 87 EB FD 73 01 FD 72 02 FD 75 03 FD 74 :8A5
8980 04 FD 23 FD 23 FD 23 FD 23 C9 22 FD 66 00 FD 6E :83D
8990 FF FD 56 FE FD 5E FD 23 FD 74 00 FD 75 FF AF ED :B49
89A0 52 38 09 FD 2B FD 2B FD 2B FD 2B C9 0B 0B 0B CD :6EA
89B0 A8 83 FE 21 C8 FE FF 20 F4 C3 20 89 23 FD 66 00 :915
89C0 FD 6E FF 22 00 A1 CD 5A 83 C9 24 FD 66 FE FD 6E :990
89D0 FD 18 F0 25 CD 77 85 3E F4 12 21 10 00 19 22 0E :5B1
89E0 A1 C9 26 CD 77 85 ED 53 0A A1 11 68 85 CD 2C 80 :7BB
89F0 11 00 A0 CD AD 81 CD 24 82 ED 5B 0A A1 3E F5 12 :757

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8A00 D5 DD 23 DD 36 00 0D DD 23 DD 36 00 36 DD 23 DD :71B
8A10 36 00 31 CD B9 87 CD E7 82 2A 00 A1 D1 19 22 0E :68F
8A20 A1 C9 27 11 00 A0 2A 02 A1 7E 12 FE 20 38 04 23 :51C
8A30 13 18 F6 CD 24 82 C9 28 2A 02 A1 DD 7E 00 77 FE :722
8A40 20 DD 2B D8 23 18 F4 29 CD 32 83 CD 5A 83 C9 2A :777
8A50 3E 0C CD A6 40 C9 2B 3E 0B CD A6 40 C9 FF 5B 0D :71D
8A60 44 84 2E 0D 0D 85 53 50 43 0D 22 85 4C 46 0D 27 :3F5
8A70 85 56 41 52 0D 2C 85 41 52 59 0D 3A 85 53 54 4B :4D6
8A80 0D A2 85 4C 49 53 54 0D F9 85 44 0D 98 86 53 44 :601
8A90 0D A0 86 24 0D A8 86 53 24 0D B0 86 3F 0D B8 86 :5D6
8AA0 40 0D C5 86 21 0D D4 86 43 4C 52 0D E9 86 44 55 :616
8AB0 50 0D EF 86 53 57 41 50 0D 0B 87 44 52 4F 50 0D :4EE
8AC0 5D 87 2B 0D 8B 87 2D 0D 97 87 2A 0D B8 87 2F 0D :538
8AD0 0E 88 3D 0D B5 88 3E 0D E2 88 3C 0D F4 88 49 46 :626
8AE0 0D 05 89 45 4C 53 45 0D 30 89 54 48 45 4E 0D 3F :405
8AF0 89 46 2E 49 46 0D 41 89 45 4E 44 0D 51 89 42 45 :4A8

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
8B00 47 49 4E 0D 6E 89 44 4F 0D 70 89 4C 4F 4F 50 0D :4C2
8B10 8A 89 49 0D BC 89 4A 0D CA 89 43 56 41 52 0D D3 :664
8B20 89 43 41 52 59 0D E2 89 43 40 0D 22 8A 43 21 0D :4DD
8B30 37 8A 3B 0D 47 8A 43 48 4D 0D 4F 8A 48 4D 0D 56 :490
8B40 8A 4E 0D 00 D0 46 52 4F 4D 0D 03 D0 42 59 0D 14 :485
```

チェック・サム プログラム・リスト

```
0 CLEAR300,&HCA00
10 DEFFNCH$(E,P)=STRING$(E-LEN(HEX$(P)),"0")+HEX$(P)
20 INPUT"Start add";S$:S=VAL("&H"+LEFT$(S$,LEN(S$)-1)+"0")
30 INPUT"End   add";E$:E=VAL("&H"+E$)
40 FOR A=S TO E STEP 16: SUM=0 : IF RIGHT$(HEX$(A),2)="00" OR A=S THEN GOSUB90
50 LPRINT FNCH$(4,A);" ";
60 FOR I=0 TO 15 : D=PEEK(I+A) :LPRINT FNCH$(2,D);" ";:SUM=SUM+D:NEXT
70 LPRINT":";FNCH$(3,SUM):NEXT
80 END
90 LPRINT:LPRINT"Add  ";:FOR I=0 TO 15:LPRINT"+";HEX$(I);" ";:NEXT:LPRINT" Sum":
RETURN
```

MZ⇨PC変換プログラム

大垣泰二

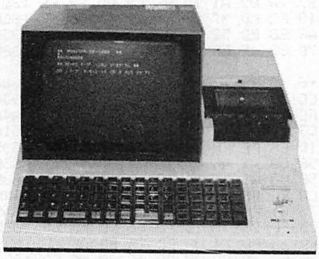




写真1　MZで作ったカセット・テープが，即PCにかけられるようになる．



最近，近くに住む友人がPC-8001を買いました．この友人はオセロが大変好きで，しかも強いのです．そこで，我がMZで走っている強いオセロ・プログラム（機械語）をPCに移植しようと思い，考え始めたのが本プログラムです．

MZとPCはCPUが同じZ80ですから，I/O以外のプログラム本体のアルゴリズムは共通に使え，プログラムがアセンブラで書かれていれば比較的簡単に移植できます．

なお，このプログラムはMZのシステム・プログラム（エディタ，アセンブラ，etc.）を持っていない人には意味がありません．機械語で打ち込むのなら直接PCに打ち込んだ方が速いからです．

プログラムの役割

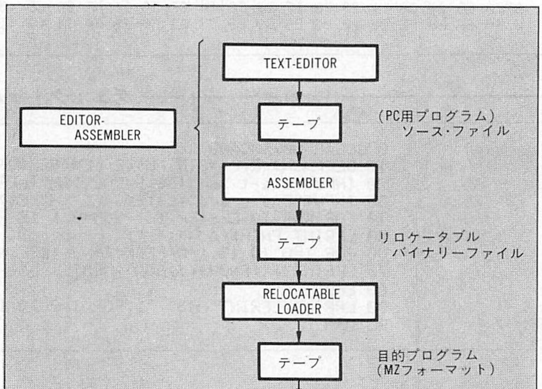
このプログラムは図1のように，MZのシステム・プログラムで開発されたPC用のMZ機械語プログラム・テープをソフトウェアだけで，正確には，外付けハードウェアなしの標準装備だけで，PCが直接読み込めるテープに変換するためのプログラムです．

もちろん元になるプログラムは，PCのハード，ソフト，つまりV-RAMエリア，キーボード・サブルーチンなどのシステム・サブルーチン，その他を充分考慮して作られていなければなりません．

アセンブルの各過程はあたかもMZのテープが直接PCのカセットレコーダからロードできるようなつもりで行ない，最終的に目的プログラムを作成します．ただし，PCの機械語プログラムはMZ上で（当然のことながら）デバッグすることができません．

そこで，PC用とMZ用の同じ働きをするI/Oサブルーチンなどをブロック化しておき，MZ用のプログラムを充分デバッグした上でこのI/Oサブルーチン・ブロックをPC用のに取り変えるのが一番確実で良い方法だと思います．
さて，このようにして得られた目的プログラムは，もちろんこのままでは記録方式もデータ・フォーマットも異なるのでPCは読み込んでくれません．そこで本プログラムの登場です．このプログラムを通すことにより，MZのテープは同じ内容のPC用機械語テープに変換され，PCにロードすることができます．

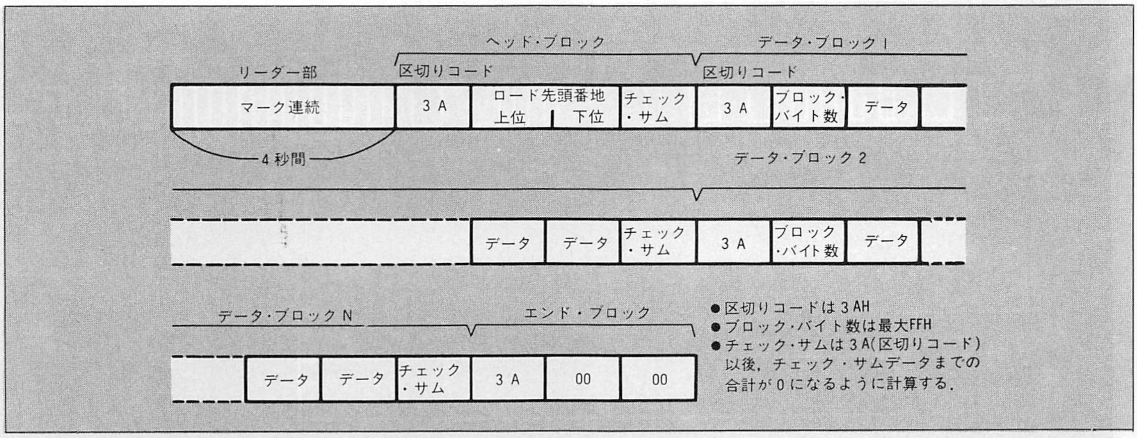
図1　MZのシステム・プログラムと本プログラムでPC用テープを作るまで．



PCのテープについて

PCの機械語テープについて，資料と実験でわかったことを説明します．

図2　PCテープのフォーマット



テープに記録されている信号は，
❶記録方式はFSK，スピードは600ボー
❷スペースは1,200Hz2サイクル
❸マークは2,400Hz4サイクル
❹データは8ビット，下位のビットから送出される．
❺スタート・ビットは1スペース
❻ストップ・ビットは1または2マーク(実験は2マークで行なったので，どちらかは不明)
のようになっており，いわゆる倍速のカンサスシティ・スタンダードです．

データのフォーマットは図2のように，
❶リーダ部（マークの連続4秒間）
❷ヘッド・ブロック
❸データ・ブロック（1個以上）
❹エンド・ブロック
の4つの部分からなり，各ブロックは区切りコード3AHで区切られています．

ヘッド・ブロックはメモリのロード・エリア先頭番地が上位8ビット，下位8ビットの順に書かれ，最後に前2バイトとの合計が00H（桁上がりは無視）になるように，チェックサム・データ1バイトがつけられます．

次のデータ・ブロックは区切りコードの直後にまず，1バイトのブロック・バイト数データが書かれます．これは，そのブロックに書かれた正味データのバイト数を表わし，最小01H～最大FFHになります．

続いて正味のデータがブロック・バイト数だけ書かれ，最後にブロック・バイト数データを含めた，そのブロックの全データとの合計が00H（同じく桁上がり無視）になるようにチェックサム・データ1バイトがつけられます．

セーブされたデータがFFHバイトまでならこれだけですが，それ以上の場合はこの後に必要な回数だけこのデータ・ブロックが繰り返されます．

最後のエンド・ブロックは単に3AH，00H，00Hと3バイト書かれているだけです．

1バイト・テープライト・サブルーチン

以上のことがらを満足させるテープを作れば，PCは読み込んでくれますが，これを外付けハードなしでやるのはなかなか大変です．

MZのハードとソフトを良く調べた人はわかると思いま

図3　600ボー信号1/2サイクルの時間とクロック数

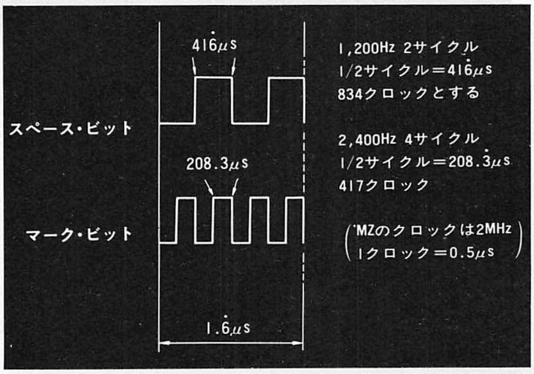


写真2　MZ用オセロ・ゲームをPCに移植し，対戦させた例

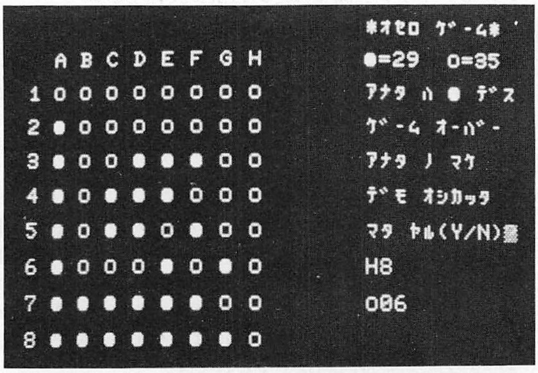


写真3　写真2はMZ，こちらはPCです．結果は……

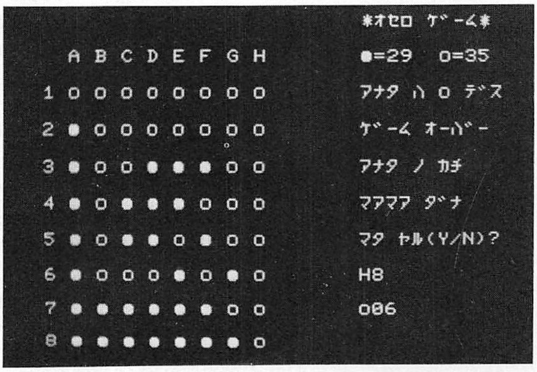
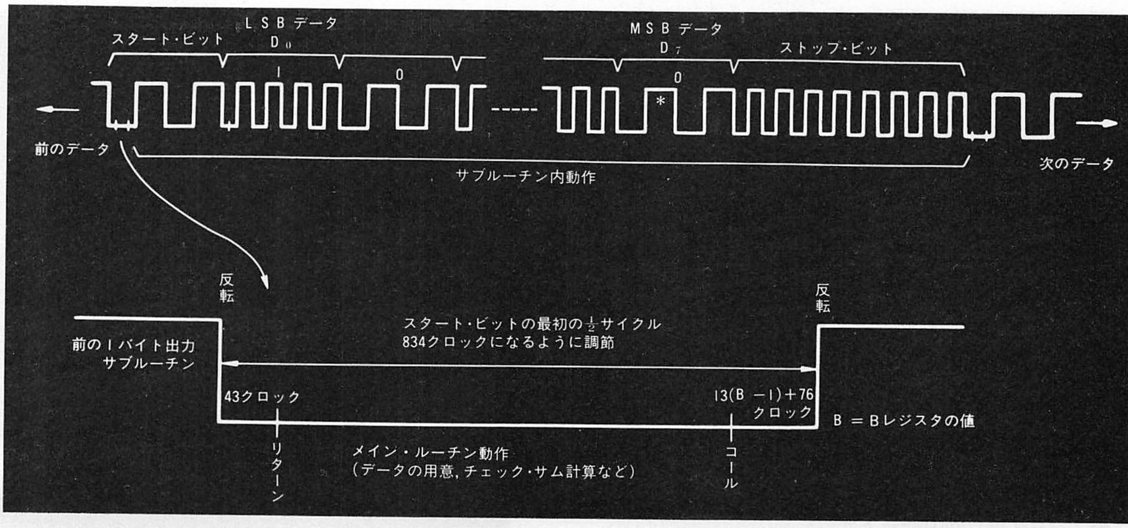


図6　1バイト テープライト・サブルーチンの動作波形



すが，**MZ**の内蔵カセット・テレコのREC入力は**8255**のPC1（ポートCのビット1）に接続されています．したがって，このPC1の出力を**図3**のようなクロック間隔で反転させてやれば，必要な信号が得られることになります．

本プログラムの中心となる，1バイト・テープライト・サブルーチンを使って，この辺のところを説明してみましょう．**図4～図6**を見ながら読んでください．

このサブルーチンが呼ばれる前に，A'，B，Cレジスタの3つが正しくセットされていなければなりません．A'レジスタ（Aレジスタの裏レジスタ）には02Hまたは03H．これはリーダー部の最後にセットされたものが，そのまま残っています．Cレジスタには出力するデータ，Bレジスタにはタイミング調整のための値を入れます．

サブルーチンがCALLされると表裏のレジスタが交換されますが，BCレジスタは同じ値にセットされます．次に，スタート・ビットのサイクル・カウンタが4にセットされます．ここで注意したいのは，このカウンタが1サイクルごとに減るのではなく，½サイクルごとに－1することです．

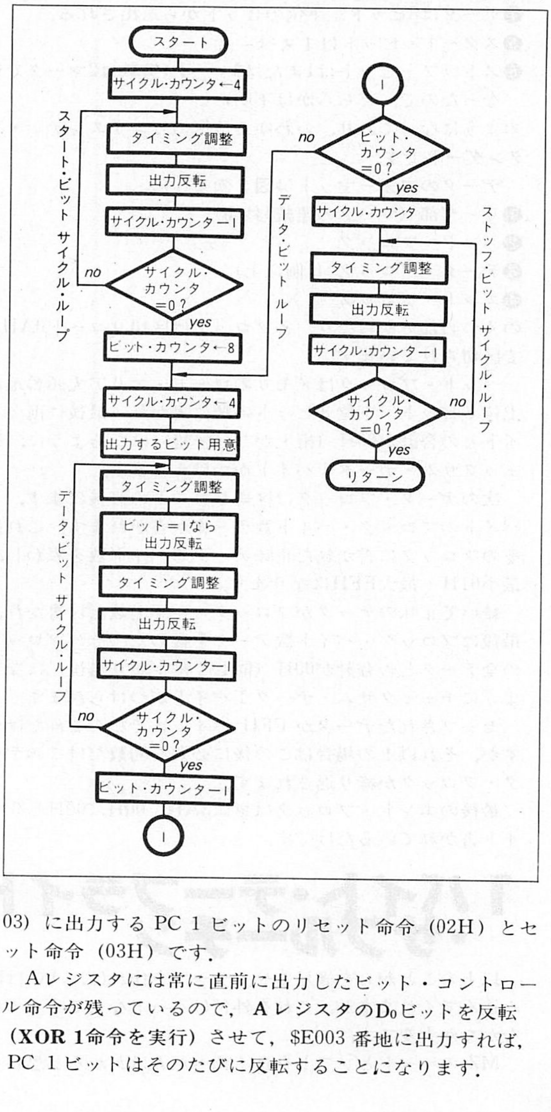
次にタイミング調整のための小ループに入りますが，ここで先にセットされたBレジスタの値が効いてきます．**図6**でわかるとおり，サブルーチンがスタートするのは時間的にスタート・ビットの最初の½サイクルの中になるので，このサブルーチンでの最初の出力反転は直前の出力反転から数えて834クロック後でなければなりません．そこで，この小ループで時間調整するわけです．

DJNZ命令はB←B－1の後，B＝0でジャンプするとき13クロック，B＝0でジャンプしないとき8クロックを使うので，自分自身にジャンプする命令では，13（B－1）＋8のクロックを消費します．したがって，このサブルーチンがCALLされてから最初の出力反転までは，他の命令のクロック数と合わせて13（B－1）＋76クロックとなり，Bレジスタの値で13クロックごとの調整ができます．

以下，サブルーチン内ではBレジスタはタイミング調整だけに使い，他の目的には使いません．13クロック以下のタイミング調整にはプログラム上意味のない命令を使用しています（図5の中で『タイミング』と注記してあるもの）．

さて，次に出力反転の方法ですが，Aレジスタに入っている02Hまたは03Hは**8255**のコントロール・ポート（$E003）に出力するPC1ビットのリセット命令（02H）とセット命令（03H）です．

図4　1バイト テープライト・サブルーチンのフローチャート



Aレジスタには常に直前に出力したビット・コントロール命令が残っているので，**A**レジスタのD_0ビットを反転（**XOR 1**命令を実行）させて，$E003番地に出力すれば，PC1ビットはそのたびに反転することになります．

図5　1バイト テープライト・サブルーチンのクロック計算

ラベル	プログラム	基本クロック数	クロック計算				ループ
	EX AF, AF"	④	4				
	PUSH BC	⑪	11				
	EXX	④	4				
	POP BC	⑩	10				
	LD D, 4	⑦	7				
TLOOP1	DJNZ TLOOP1	⑬ ↓⑧	注1)	775 注2)			スタート・ビット サイクル・ループ
	JR +2 ｝タイミング	⑫	12	12			
	XOR 1	⑦	7	7			
	LD (E003H), A	⑬	13	13			
出力反転							
	NOP ｝タイミング	④	4	4			
	LD B, 60	⑦	7	7			
	DEC D	④	4	4			
	JR NZ, TLOOP1	⑫ ↓⑦	7	12		T834クロック	
	LD IX, 0 ｝タイミング	⑭	14				
	LD B, 25	⑦	7				
	LD L, 78	⑦	7				
DTLOOP	LD D, 4	⑦	7	7			データ・ビット ループ
	LD E, 0	⑦	7	7			
	RRC C	⑧	8	8			
	RL E	⑧	8	8			
TLOOP2	DJNZ TLOOP2	⑬ ↓⑧	320 注3)	307 注4)	359 注5)		データ・ビット サイクル・ループ
	XOR E	④	4	4	4		
ビット=1なら出力反転	LD (E003H), A	⑬	13	13	13		
	LD I, A ｝タイミング	⑨		9			
	LD I, A	⑨		9			
	LD B, 29	⑦		7			
TLOOP3	DJNZ TLOOP3	⑬ ↓⑧	372 (13×28+8)				
	XOR 1	⑦		7			
	LD (E003H), A	⑬		13		T147クロック	
出力反転							
	LD I, A ｝タイミング	⑨	9	9	9		
	LD I, A	⑨	9	9	9		
	LD B, 28	⑦	7	7	7		
	DEC D	④	4	4	4		
	JR NZ, TLOOP2	⑫ ↓⑦	7	7	12	T417クロック	
	NOP ｝タイミング	④	4	4			
	LD B, 24	⑦	7	7			
	DEC L	④	4	4			
	JR NZ, DTLOOP	⑫ ↓⑦	7	12		T417クロック	
	INC B	④	4				
	LD D, 16	⑦	7				
TLOOP4	DJNZ TLOOP4	⑬ ↓⑧	320 注6)	359 注7)			ストップ・ビット サイクル・ループ
	NOP ｝タイミング	④	4	4			
	NOP	④	4	4			
	XOR 1	⑦	7	7			
	LD (E003H), A	⑬	13	13			
出力反転							
	LD H, 0 ｝タイミング	⑦	7	7			
	LD B, 28	⑦	7	7			
	DEC D	④	4	4			
	JR NZ, TLOOP4	⑫ ↓⑦	7	12		T417クロック	
	EXX	④	4	OUT			
	EX AF, AF'	④	4	T43クロック			
	RET	⑩	10				

- A'レジスタにはPC1のコントロール・コードが入っている.
- 引き数C=データ，B=タイミング・カウント
- 13(B−1) +76クロック，前の出力反転から数えて834クロックに合わせる.

注1) 13(B−1)+8　注2) 13×59+8　注3) 13×24+8　注4) 13×23+8　注5) 13×27+8　注6) 13×24+8　注7) 13×27+8

このようにしてタイミングを取りながら，834クロックごとの出力反転を4回繰り返せば，スタート・ビット（1200Hz2サイクル）ができ上がります.

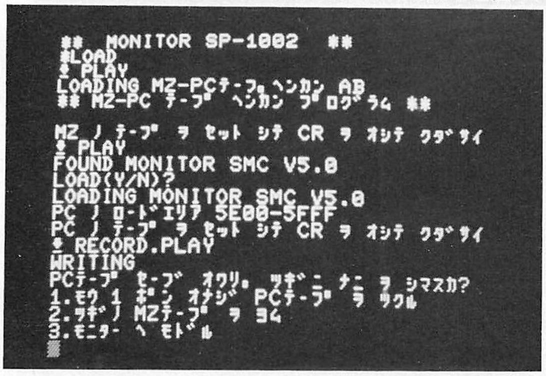
次のデータ・ビット出力でも，同じようにタイミング調整しながら出力反転を繰り返せば良いのですが，ここではデータ・ビットの値で反転の数が2倍になるので，834クロックごとの出力反転のちょうど中間に，出力するビットが1のとき反転し，0のとき反転しない命令を入れておきます.

最後に417クロックの出力反転を16回繰り返しストップ・ビット×2を作ります.

なお，最後の出力反転の後リターンするまでに43クロックがあるので，これは次のタイミング計算のとき，加算しなければなりません.



写真4　変換プログラムでPC用テープを作る過程



プログラムの説明

さてやっとプログラム全体の説明ですが，ここまでくれば話は簡単です．図7のフローチャートのように，MZのテープをバッファ・エリアに読み込んで，PCフォーマットでセーブすれば良いのです．他のことは，このプログラムを使いやすくするための工夫です．

詳しくは，最後に載っているプログラム・リストを読んでください．プログラムのスタートは$3000番地です．なお，リロケートする場合，変更するワーキング・エリアは前の方にまとめてあります．この部分だけ変えれば充分です．

MZテープのフォーマットについてはまだわからないことが多いので，詳しくは書きません．プログラムどおりに使ってください．PCテープ・ラインについてはフローチャートにしておいたので前出のPCテープ説明と共に読んでください．なお，プログラム・リスト中でもPCテープライト中はBレジスタはタイミング調整だけに使います．また，『タイミング』とコメントされている命令もすべてタイミング調整用です．



使い方

最後に本プログラムの使い方を説明します．スタートさせるとまず，

```
** MZ-PC テープ ヘンカン プログラム **
MZ ノ テープ ヲ セット シテ CR ヲ オシテ クダサイ
```

と表示されるので，

MZのシステム・プログラムで作ったPC用のマシン語プログラム・テープをセットして CR を押します．すると，

```
↓ PLAY
```

と出るので，カセットのPLAYボタンを押します．セーブされているファイルを見つけると，

```
FOUND ×××××
LOAD (Y/N) ?■
```

とロードするか否か聞いてきます．このとき， N を押せば次のファイルを捜し， Y を押せば本体のデータをロードし始め，

```
LOADING ×××××
```

のメッセージがでます．ロードが終わると，

```
PC ノ ロードエリア △△△△−○○○○
PC ノ テープ ヲ セット シテ CR ヲ オシテ クダサイ.
```

と表示されます．PCのロード・アドレスはリロケータブル・ローダで目的プログラムをセーブするときにきまってしまい，本プログラムではその情報をコピーするだけで変

図7　全体のフローチャート

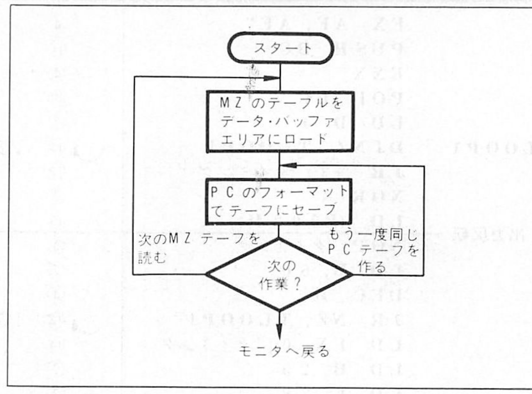
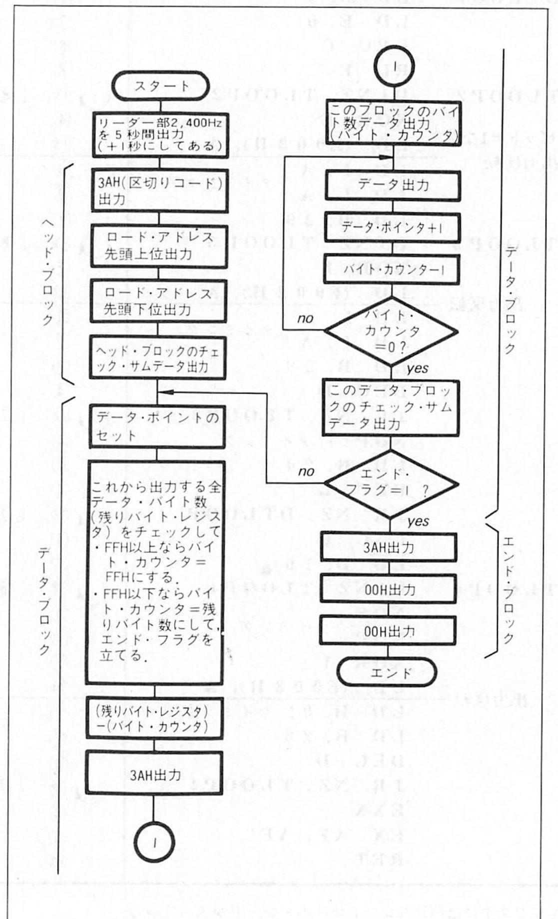


図8　PCフォーマット・テープ(ライト)のフローチャート



えることはできません.

さて PC にロードするためのカセットを MZ のカセット・レコーダにセットして **CR** を押すと,

```
↓RECORD.PLAY
```

のメッセージが出るので, レコードと PLAY ボタンを押すと,

```
WRITING
```

と PC フォーマットでセーブが始まります. 終わると,

```
PCﾃｯﾌﾟ ｾｰﾌﾞ ｵﾜﾘ. ﾂｷﾞﾆ ﾅﾆｦｼﾏｽｶ?
1. ﾓｳ1ﾎﾟﾝ ｵﾅｼﾞPCﾃｰﾌﾟ ｦ ﾂｸﾙ
2. ﾂｷﾞﾉ MZﾃｰﾌﾟ ｦ ﾖﾑ
3. ﾓﾆﾀｰ ﾍ ﾓﾄﾞﾙ
```

と出るので, 1, 2, 3の内1つを選んで数字のキーを押してください.

あとがき

本プログラムはまだできたてで, 充分に使い込んでいないので, まだ不備な点があるかもしれません. 何か気がついたことがあったら, I/O編集部まで一報ください.

今回は, まだ PC テープの読み込みが ALL ソフト化していないので (FSKインターフェイスを外付けした場合のプログラムは走っています), 書き込んだPCテープをベリファイできません. 次にはこれに挑戦し, さらに将来はMZ↔PCの BASIC テキスト変換プログラムにも挑戦してみたいと思います.



MZ→PCテープ変換　プログラム・リスト

```
0000                    REL    3000H
3000
3000                  ;****************************************
3000                  ;*     MZ-PC ﾃｰﾌﾟﾍﾝｶﾝ ﾌﾟﾛｸﾞﾗﾑ V1.0    *
3000                  ;****************************************
3000                  ;
3000                  ;***** ﾜｰｷﾝｸﾞ ｴﾘｱ ********************
3000                  ;
3000  P               HDADDR: EQU   2F00H        ; ｾﾝﾄｳ ﾊﾞﾝﾁ ﾚｼﾞｽﾀ
3000  P               ALLBYT: EQU   2F02H        ; ﾃﾞｰﾀｰ ﾊﾞｲﾄ ｽｳ ﾚｼﾞｽﾀ
3000  P               LSTBYT: EQU   2F04H        ; ﾉｺﾘ ﾊﾞｲﾄ ｽｳ ﾚｼﾞｽﾀ
3000  P               DATABF: EQU   4000H        ; ﾃﾞｰﾀｰ ﾊﾞｯﾌｧ ｴﾘｱ
3000                  ;
3000                  ;***** ﾒｲﾝ ﾙｰﾁﾝ ********************
3000                  ;
3000                  ;*     ﾀｲﾄﾙ ﾌﾟﾘﾝﾄ       *
3000 CD0600           MSTRT:  CALL  0006H        ; CR
3003 110D32                   LD    DE,MSG0      ; ﾀｲﾄﾙ
3006 CD1500                   CALL  0015H        ; ﾌﾟﾘﾝﾄ
3009 CD0600                   CALL  0006H        ; CR
300C                  ;*     MZﾃｰﾌﾟ ﾘｰﾄﾞ      *
300C CD0600           RDSTRT: CALL  0006H        ; CR
300F 112B32                   LD    DE,MSG1      ; MZﾃｰﾌﾟ ﾒｯｾｰｼﾞ
3012 CD1500                   CALL  0015H        ; ﾌﾟﾘﾝﾄ
3015 CD3E00                   CALL  003EH        ; BELL
3018 CDB309                   CALL  09B3H        ; CR ﾆｭｳﾘｮｸ ﾏﾁ
301B CD2700           HBLKRD: CALL  0027H        ; ﾍｯﾄﾞﾌﾞﾛｯｸ ﾘｰﾄﾞ
301E 384B                     JR    C,RDEROR     ; ﾘｰﾄﾞ ｴﾗｰ
3020 CD0600                   CALL  0006H        ; CR
3023 113101                   LD    DE,0131H     ; FOUND ﾒｯｾｰｼﾞ
3026 CD1500                   CALL  0015H        ; ﾌﾟﾘﾝﾄ
3029 11F110                   LD    DE,10F1H     ; ﾌｧｲﾙﾈｰﾑ
302C CD1500                   CALL  0015H        ; ﾌﾟﾘﾝﾄ
302F CD0600                   CALL  0006H        ; CR
3032 114D32                   LD    DE,MSG2      ; LOAD(Y/N)? ﾒｯｾｰｼﾞ
3035 CD1500                   CALL  0015H        ; ﾌﾟﾘﾝﾄ
3038 CD3E00                   CALL  003EH        ; BELL
303B CDB309                   CALL  09B3H        ; ｷｰ ﾆｭｳﾘｮｸ ﾏﾁ
303E FE19                     CP    19H          ; "Y"?
3040 20D9                     JR    NZ,HBLKRD    ; "Y" ｲｶﾞｲ ﾅﾗ ﾍｯﾄﾞﾌﾞﾛｯｸ
ﾘｰﾄﾞ ﾍ ﾓﾄﾞﾙ
3042 CD0600                   CALL  0006H        ; CR
3045 113801                   LD    DE,0138H     ; LOADING ﾒｯｾｰｼﾞ
3048 CD1500                   CALL  0015H        ; ﾌﾟﾘﾝﾄ
304B 11F110                   LD    DE,10F1H     ; ﾌｧｲﾙﾈｰﾑ
304E CD1500                   CALL  0015H        ; ﾌﾟﾘﾝﾄ
3051 CD0600                   CALL  0006H        ; CR
3054 2A0411                   LD    HL,(1104H)   ; PC ﾛｰﾄﾞｱﾄﾞﾚｽ ｾｰﾌﾞ
3057 22002F                   LD    (HDADDR),HL
305A 210040                   LD    HL,DATABF    ; ﾃﾞｰﾀｰﾊﾞｯﾌｧ ｴﾘｱ ｾｯﾄ
305D 220411                   LD    (1104H),HL
3060 2A0211                   LD    HL,(1102H)   ; ﾃﾞｰﾀｰ ﾊﾞｲﾄ ｽｳ ｹﾞｯﾄ
3063 22022F                   LD    (ALLBYT),HL
3066 CD2A00                   CALL  002AH        ; ﾃﾞｰﾀｰﾌﾞﾛｯｸ ﾘｰﾄﾞ
3069 3014                     JR    NC,WRINIS    ; ｴﾗｰ ﾅｼ PCﾃｰﾌﾟ ﾏｴｼｮﾘ ﾍ
306B                  ;
306B                  ;*     ｴﾗｰ ｼｮﾘ        *
306B CD0600           RDEROR: CALL  0006H
306E 11B501                   LD    DE,01B5H     ; ｴﾗｰ ﾒｯｾｰｼﾞ
3071 CD1500                   CALL  0015H        ; ﾌﾟﾘﾝﾄ
3074 CD3E00                   CALL  003EH        ; BELL
3077 CD3E00                   CALL  003EH
```

MZ→PCテープ変換　プログラム・リスト

```
307A CD3E00              CALL  003EH
307D 188D                JR    RDSTRT        ;MZﾃｰﾌﾟ ﾘｰﾄﾞ ﾔﾘﾅｵｼ
307F             ;
307F             ;*      PCﾃｰﾌﾟ ﾏｴｼｮﾘ     *
307F 115832      WRINIS: LD    DE,MSG3       ; PCﾛｰﾄﾞ ｴﾘｱ ﾒｯｾｰｼﾞ
3082 CD1500              CALL  0015H         ; ﾌﾟﾘﾝﾄ
3085 2A002F              LD    HL,(HDADDR)   ; ﾛｰﾄﾞ ｾﾝﾄｳ ﾊﾞﾝﾁ
3088 CDBA03              CALL  03BAH         ; 4ｹﾀ HEX ﾌﾟﾘﾝﾄ
308B 3E2D                LD    A,2DH         ; "-"
308D CD1200              CALL  0012H         ; 1ﾓｼﾞ ﾌﾟﾘﾝﾄ
3090 EB                  EX    DE,HL
3091 2A022F              LD    HL,(ALLBYT)   ; ﾛｰﾄﾞ ﾊﾞｲﾄ ｽｳ
3094 22042F              LD    (LSTBYT),HL   ; ﾉｺﾘﾊﾞｲﾄ ﾚｼﾞｽﾀ ｾｯﾄ
3097 19                  ADD   HL,DE         ; ｻｲｼｭｳ ﾊﾞﾝﾁ ｹｲｻﾝ
3098 2B                  DEC   HL
3099 CDBA03              CALL  03BAH         ; 4ｹﾀ HEX ﾌﾟﾘﾝﾄ
309C             ;*      PCﾃｰﾌﾟ ﾗｲﾄ ｽﾀｰﾄ     *
309C CD0600      WRSTRT: CALL  0006H         ; CR
309F 116632              LD    DE,MSG4       ; PCﾃｰﾌﾟ ﾒｯｾｰｼﾞ
30A2 CD1500              CALL  0015H         ; ﾌﾟﾘﾝﾄ
30A5 CD3E00              CALL  003E          ; BELL
30A8 CDB309              CALL  09B3          ; CR ﾆｭｳﾘｮｸ ﾏﾁ
30AB CD0600              CALL  0006H         ; CR
30AE 112907              LD    DE,0729H      ; ﾚｺｰﾄﾞﾎﾞﾀﾝ ﾒｯｾｰｼﾞ
30B1 CD1800              CALL  0018H         ; ﾌﾟﾘﾝﾄ
30B4 112407              LD    DE,0724H      ; ﾌﾟﾚｲﾎﾞﾀﾝ ﾒｯｾｰｼﾞ
30B7 CD1800              CALL  0018H         ; ﾌﾟﾘﾝﾄ
30BA 3A02E0      PLYCH:  LD    A,(E002H)     ; ﾌﾟﾚｲﾎﾞﾀﾝ ﾁｪｯｸ
30BD E610                AND   10H
30BF 28F9                JR    Z,PLYCH       ; ﾌﾟﾚｲﾎﾞﾀﾝ ON ﾏﾁ
30C1 CD0600              CALL  0006H         ; CR
30C4 116C04              LD    DE,046CH      ; ﾗｲﾃｨﾝｸﾞ ﾒｯｾｰｼﾞ
30C7 CD1500              CALL  0015H         ; ﾌﾟﾘﾝﾄ
30CA             ;*      ﾘｰﾀﾞｰ ﾌﾞ ﾗｲﾄ     *
30CA 3E02                LD    A,2           ; 8255 ｺﾝﾄﾛｰﾙ ｺｰﾄﾞ ｲﾆｼｬﾙ
30CC 21AC5D              LD    HL,5DACH      ; ｻｲｸﾙ ｶｳﾝﾀ ｾｯﾄ
30CF 11FFFF              LD    DE,FFFFH      ; -1 ｾｯﾄ
30D2 061C        RDLOOP: LD    B,28
30D4 10FE        TLOOP0: DJNZ  TLOOP0        ; ﾀｲﾐﾝｸﾞ ﾙｰﾌﾟ0
30D6 00                  NOP                 ;ﾀｲﾐﾝｸﾞ
30D7 00                  NOP                 ;ﾀｲﾐﾝｸﾞ
30D8 EE01                XOR   1
30DA 3203E0              LD    (E003H),A     ; ｼｭﾂﾘｮｸ ﾊﾝﾃﾝ
30DD 19                  ADD   HL,DE         ; (ｻｲｸﾙ ｶｳﾝﾀ)-1
30DE 38F2                JR    C,RDLOOP
30E0 08                  EX    AF,AF'        ; ｺﾝﾄﾛｰﾙ ｺｰﾄﾞ ﾎｿﾞﾝ
30E1             ;*      ﾍｯﾀﾞ ﾌﾞﾛｯｸ ﾗｲﾄ     *
30E1 2A002F              LD    HL,(HDADDR)   ; ﾛｰﾄﾞ ｾﾝﾄｳ ﾊﾞﾝﾁ
30E4 AF                  XOR   A
30E5 0E3A                LD    C,3AH
30E7 0635                LD    B,53
30E9 ED47                LD    I,A           ; ﾀｲﾐﾝｸﾞ
30EB CDB031              CALL  1BYTWR        ; 3AH ﾗｲﾄ
30EE 4C                  LD    C,H
30EF 91                  SUB   C             ; ﾁｪｯｸｻﾑ ｹｲｻﾝ
30F0 0600                LD    B,0           ; ﾀｲﾐﾝｸﾞ
30F2 0635                LD    B,53
30F4 CDB031              CALL  1BYTWR        ; ﾛｰﾄﾞｱﾄﾞﾚｽ H ﾗｲﾄ
30F7 4D                  LD    C,L
30F8 91                  SUB   C             ; ﾁｪｯｸｻﾑ ｹｲｻﾝ
30F9 0600                LD    B,0           ; ﾀｲﾐﾝｸﾞ
30FB 0635                LD    B,53
30FD CDB031              CALL  1BYTWR        ; ﾛｰﾄﾞｱﾄﾞﾚｽ L ﾗｲﾄ
3100 4F                  LD    C,A
3101 00                  NOP                 ;ﾀｲﾐﾝｸﾞ
3102 0600                LD    B,0           ; ﾀｲﾐﾝｸﾞ
3104 0635                LD    B,53
3106 CDB031              CALL  1BYTWR        ; ﾁｪｯｸｻﾑ ﾗｲﾄ
3109             ;*      ﾃﾞｰﾀｰ ﾌﾞﾛｯｸ ﾗｲﾄ     *
3109 210000              LD    HL,0          ; ﾀｲﾐﾝｸﾞ
310C 210040              LD    HL,DATABF     ; ﾊﾞｯﾌｧｰ ｴﾘｱ ｾﾝﾄｳ ﾊﾞﾝﾁ
310F E5          DTBLCK: PUSH  HL            ; ｹｲｻﾝ ﾉ ﾀﾒ ｾｰﾌﾞ
3110 AF                  XOR   A
3111 2A042F              LD    HL,(LSTBYT)   ; ﾉｺﾘ ﾊﾞｲﾄ ｽｳ ｦ ｼﾗﾍﾞﾙ
3114 11FF00              LD    DE,00FFH
3117 ED52                SBC   HL,DE
3119 062C                LD    B,44
311B 00                  NOP                 ;ﾀｲﾐﾝｸﾞ
311C 2808                JR    Z,ENDFG       ; LSTBYT=FFH ﾅﾗ ｴﾝﾄﾞﾌﾗｸﾞ
 ｦ ﾀﾃﾙ
311E 3008                JR    NC,DBSTRT     ; LSTBYT>FFH ﾅﾗ ﾃﾞｰﾀｰﾌﾞﾛ
ｯｸ ｽﾀｰﾄ ﾍ
3120 19                  ADD   HL,DE         ; LSTBYT<FFH ﾅﾗ ﾉｺﾘ ﾊﾞｲﾄ
 ｦ ｶｳﾝﾀ ﾆ ｾｯﾄ ｼﾃ ｴﾝﾄﾞﾌﾗｸﾞ ｦ ﾀﾃﾙ
3121 5D                  LD    E,L           ; ﾊﾞｲﾄｶｳﾝﾀ ｾｯﾄ
3122 DD09                ADD   IX,BC         ; ﾀｲﾐﾝｸﾞ
3124 0629                LD    B,41
3126 1601        ENDFG:  LD    D,1           ; ｴﾝﾄﾞﾌﾗｸﾞ ｦ ﾀﾃﾙ
3128 22042F      DBSTRT: LD    (LSTBYT),HL   ; ｻｼﾋｷ ﾉｺﾘ ﾊﾞｲﾄ ｦ ﾓﾄﾞｽ
```

```
312B E1                        POP   HL
312C 0E3A                      LD    C,3AH
312E CDB031                    CALL  1BYTWR              ; 3AH ﾗｲﾄ
3131 4B                        LD    C,E
3132 91                        SUB   C                   ; ﾁｪｯｸｻﾑ ｹｲｻﾝ
3133 0600                      LD    B,0                 ; ﾀｲﾐﾝｸﾞ
3135 0635                      LD    B,53
3137 CDB031                    CALL  1BYTWR              ; ﾌﾞﾛｯｸ ﾊﾞｲﾄｽｳ ﾗｲﾄ
313A 0635                      LD    B,53
313C 4E               DBLOOP:  LD    C,(HL)              ; ﾃﾞｰﾀｰ ﾗｲﾄ ﾙｰﾌﾟ
313D 91                        SUB   C                   ; ﾁｪｯｸｻﾑ ｹｲｻﾝ
313E 00                        NOP                       ;ﾀｲﾐﾝｸﾞ
313F CDB031                    CALL  1BYTWR              ; ﾃﾞｰﾀｰ ﾗｲﾄ
3142 00                        NOP                       ;ﾀｲﾐﾝｸﾞ
3143 0633                      LD    B,51
3145 23                        INC   HL
3146 1D                        DEC   E
3147 20F3                      JR    NZ,DBLOOP
3149 4F                        LD    C,A
314A ED47                      LD    I,A                 ; ﾀｲﾐﾝｸﾞ
314C 0633                      LD    B,51
314E CDB031                    CALL  1BYTWR              ; ﾁｪｯｸｻﾑ ﾗｲﾄ
3151 00                        NOP                       ;ﾀｲﾐﾝｸﾞ
3152 15                        DEC   D                   ; ｴﾝﾄﾞﾌﾗｸﾞ=1?
3153 20BA                      JR    NZ,DTBLCK           ;ｴﾝﾄﾞﾌﾗｸﾞ=0 ﾅﾗ ﾂｷﾞﾉ ﾃﾞｰ
ﾀｰﾌﾞﾛｯｸ ﾍ
3155                  ;*     ｴﾝﾄﾞﾌﾞﾛｯｸ ﾗｲﾄ      *
3155 0E3A                      LD    C,3AH
3157 0634                      LD    B,52
3159 23                        INC   HL                  ; ﾀｲﾐﾝｸﾞ
315A CDB031                    CALL  1BYTWR              ; 3AH ﾗｲﾄ
315D 0E00                      LD    C,0
315F 0635                      LD    B,53
3161 00                        NOP                       ;ﾀｲﾐﾝｸﾞ
3162 00                        NOP                       ;ﾀｲﾐﾝｸﾞ
3163 CDB031                    CALL  1BYTWR              ; 00H ﾗｲﾄ
3166 0E00                      LD    C,0
3168 0635                      LD    B,53
316A 00                        NOP                       ;ﾀｲﾐﾝｸﾞ
316B 00                        NOP                       ;ﾀｲﾐﾝｸﾞ
316C CDB031                    CALL  1BYTWR              ; 00H ﾗｲﾄ
316F                  ;*     ｱﾄｼｮﾘ      *
316F CD0007                    CALL  0700H               ; ｶｾｯﾄ ﾓｰﾀｰ ｽﾄｯﾌﾟ
3172 CD0600           PCWEND:  CALL  0006H               ; CR
3175 118832                    LD    DE,MSG5             ; PCﾃｰﾌﾟ ｵﾜﾘ ﾒｯｾｰｼﾞ
3178 CD1500                    CALL  0015H               ; ﾌﾟﾘﾝﾄ
317B CD0600                    CALL  0006H               ; CR
317E 11A932                    LD    DE,MSG6             ; PC ﾓﾄﾞﾘ ﾒｯｾｰｼﾞ
3181 CD1500                    CALL  0015H               ; ﾌﾟﾘﾝﾄ
3184 CD0600                    CALL  0006H               ; CR
3187 11C632                    LD    DE,MSG7             ; MZ ﾓﾄﾞﾘ ﾒｯｾｰｼﾞ
318A CD1500                    CALL  0015H               ; ﾌﾟﾘﾝﾄ
318D CD0600                    CALL  0006H               ;CR
3190 11D932                    LD    DE,MSG8             ;ﾓﾆﾀｰ ﾒｯｾｰｼﾞ
3193 CD1500                    CALL  0015H               ; ﾌﾟﾘﾝﾄ
3196 CD0600                    CALL  0006H               ; CR
3199 CD3E00                    CALL  003EH               ; BELL
319C CDB309                    CALL  09B3H               ; ｷｰ ﾆｭｳﾘｮｸ ﾏﾁ
319F FE21                      CP    21H                 ; "1"ﾅﾗ
31A1 CA7F30                    JP    Z,WRINIS            ; PCﾃｰﾌﾟ ﾗｲﾄ ﾍ
31A4 FE22                      CP    22H                 ; "2"ﾅﾗ
31A6 CA0C30                    JP    Z,RDSTRT            ; MZﾃｰﾌﾟ ﾘｰﾄﾞ ﾍ
31A9 FE23                      CP    23H                 ; "3"ﾅﾗ
31AB CA0000                    JP    Z,0000H             ; ﾓﾆﾀｰ ﾍ
31AE 18C2                      JR    PCWEND              ; ｷｰ ｻｲﾆｭｳﾘｮｸ
31B0                  ;
31B0                  ;
31B0                  ;***** ｻﾌﾞﾙｰﾁﾝ ********************
31B0                  ;
31B0                  ;*     1ﾊﾞｲﾄ ﾃｰﾌﾟ ﾗｲﾄ      *
31B0                  ;
31B0                  ; B=ﾀｲﾐﾝｸﾞｶｳﾝﾄ  C=ﾃﾞｰﾀ
31B0                  ;
31B0 08               1BYTWR:  EX    AF,AF'
31B1 C5                        PUSH  BC
31B2 D9                        EXX
31B3 C1                        POP   BC
31B4 1604                      LD    D,4                 ; ｽﾄｯﾌﾟﾋﾞｯﾄ ｻｲｸﾙ ｶｳﾝﾀ ｾｯ
ﾄ
31B6 10FE             TLOOP1:  DJNZ  TLOOP1              ; ﾀｲﾐﾝｸﾞ ﾙｰﾌﾟ1
31B8 1800                      JR    +2                  ; ﾀｲﾐﾝｸﾞ
31BA EE01                      XOR   1
31BC 3203E0                    LD    (E003H),A           ; ｼｭﾂﾘｮｸ ﾊﾝﾃﾝ
31BF 00                        NOP                       ;ﾀｲﾐﾝｸﾞ
31C0 063C                      LD    B,60
31C2 15                        DEC   D
31C3 20F1                      JR    NZ,TLOOP1
31C5 DD210000                  LD    IX,0                ; ﾀｲﾐﾝｸﾞ
31C9 0619                      LD    B,25
31CB 2E08                      LD    L,8                 ; ﾃﾞｰﾀ ﾋﾞｯﾄ ｶｳﾝﾀ ｾｯﾄ
```

MZ→PCテープ変換　プログラム・リスト

```
31CD 1604          DTLOOP: LD    D,4          ; ﾃﾞｰﾀ ﾋﾞｯﾄ ｻｲｸﾙ ｶｳﾝﾀ ｾｯ
ﾄ
31CF 1E00                  LD    E,0          ; ﾋﾞｯﾄ ﾊﾝﾃﾝ ﾜｰｷﾝｸﾞﾚｼﾞｽﾀ
ｸﾘｱ
31D1 CB09                  RRC   C            ; ｷｬﾘｰ←ﾃﾞｰﾀﾋﾞｯﾄ
31D3 CB13                  RL    E            ; ﾜｰｷﾝｸﾞﾚｼﾞｽﾀ←ｷｬﾘｰ
31D5 10FE          TLOOP2: DJNZ  TLOOP2       ; ﾀｲﾐﾝｸﾞ ﾙｰﾌﾟ2
31D7 AB                    XOR   E
31D8 3203E0                LD    (E003H),A    ; ﾃﾞｰﾀﾋﾞｯﾄ=1 ﾅﾗ ｼﾂﾘｮｸ ﾊﾝ
ﾃﾝ
31DB ED47                  LD    I,A          ; ﾀｲﾐﾝｸﾞ
31DD ED47                  LD    I,A          ; ﾀｲﾐﾝｸﾞ
31DF 061D                  LD    B,29
31E1 10FE          TLOOP3: DJNZ  TLOOP3       ; ﾀｲﾐﾝｸﾞ ﾙｰﾌﾟ3
31E3 EE01                  XOR   1
31E5 3203E0                LD    (E003H),A    ; ｼﾂﾘｮｸ ﾊﾝﾃﾝ
31E8 ED47                  LD    I,A          ; ﾀｲﾐﾝｸﾞ
31EA ED47                  LD    I,A          ; ﾀｲﾐﾝｸﾞ
31EC 061C                  LD    B,28
31EE 15                    DEC   D
31EF 20E4                  JR    NZ,TLOOP2
31F1 00                    NOP                ;ﾀｲﾐﾝｸﾞ
31F2 0618                  LD    B,24
31F4 2D                    DEC   L
31F5 20D6                  JR    NZ,DTLOOP
31F7 04                    INC   B
31F8 1610                  LD    D,16         ; ｽﾄｯﾌﾟﾋﾞｯﾄ ｻｲｸﾙ ｶｳﾝﾀ ｾｯ
ﾄ
31FA 10FE          TLOOP4: DJNZ  TLOOP4       ; ﾀｲﾐﾝｸﾞ ﾙｰﾌﾟ4
31FC 00                    NOP                ;ﾀｲﾐﾝｸﾞ
31FD 00                    NOP                ;ﾀｲﾐﾝｸﾞ
31FE EE01                  XOR   1
3200 3203E0                LD    (E003H),A    ; ｼﾂﾘｮｸ ﾊﾝﾃﾝ
3203 2600                  LD    H,0          ; ﾀｲﾐﾝｸﾞ
3205 061C                  LD    B,28
3207 15                    DEC   D
3208 20F0                  JR    NZ,TLOOP4
320A D9                    EXX
320B 08                    EX    AF,AF'
320C C9                    RET
320D               ;
320D               ;***** ﾒｯｾｰｼﾞ ************************
320D               ;
320D 2A2A204D      MSG0:   DEFM  '** MZ-PC ﾃｰﾌﾟ ﾍﾝｶﾝ ﾌﾟﾛｸﾞﾗﾑ **'
3211 5A2D5043
3215 20A390AC
3219 BF20ADBD
321D 96BD20AC
3221 BFBB98BE
3225 B7B1202A
3229 2A
322A 0D                    DEFB  0DH
322B 4D5A20A9      MSG1:   DEFM  'MZ ﾉ ﾃｰﾌﾟ ｦ ｾｯﾄ ｼﾃ CR ｦ ｵｼﾃ ｸﾀﾞｻｲ'
322F 20A390AC
3233 BF208620
3237 9E8FA420
323B 9CA32043
323F 52208620
3243 959CA320
3247 98A0BE9B
324B 92
324C 0D                    DEFB  0DH
324D 4C4F4144      MSG2:   DEFM  'LOAD(Y/N)?'
3251 28592F4E
3255 293F
3257 0D                    DEFB  0DH
3258 504320A9      MSG3:   DEFM  'PC ﾉ ﾛｰﾄﾞｴﾘｱ '
325C 20BB90A4
3260 BE94B891
3264 20
3265 0D                    DEFB  0DH
3266 504320A9      MSG4:   DEFM  'PC ﾉ ﾃｰﾌﾟ ｦ ｾｯﾄ ｼﾃ CR ｦ ｵｼﾃ ｸﾀﾞｻｲ'
326A 20A390AC
326E BF208620
3272 9E8FA420
3276 9CA32043
327A 52208620
327E 959CA320
3282 98A0BE9B
3286 92
3287 0D                    DEFB  0DH
3288 5043A390      MSG5:   DEFM  'PCﾃｰﾌﾟ ｾｰﾌﾞ ｵﾜﾘ｡ ﾂｷﾞﾆ ﾅﾆ ｦ ｼﾏｽｶ?'
328C ACBF209E
3290 90ACBE20
3294 95BCB881
3298 20A297BE
329C A620A5A6
32A0 2086209C
32A4 AF9D963F
```

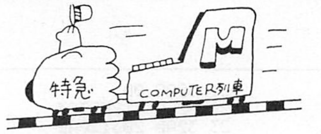
```
32A8 0D                       DEFB   0DH
32A9 312EB393       MSG6:     DEFM   '1.ﾓｳ 1 ﾎﾟﾝ ｵﾅｼﾞ PCﾃｰﾌﾟ ｦ ﾂｸﾙ'
32AD 203120AE
32B1 BFBD2095
32B5 A59CBE20
32B9 5043A390
32BD ACBF2086
32C1 20A298B9
32C5 0D                       DEFB   0DH
32C6 B22EA297       MSG7:     DEFM   '2.ﾂｷﾞﾉ MZﾃｰﾌﾟ ｦ ﾖﾑ'
32CA BEA9204D
32CE 5AA390AC
32D2 BF208620
32D6 B6B1
32D8 0D                       DEFB   0DH
32D9 332EB3A6       MSG8:     DEFM   '3.ﾓﾆﾀｰ ﾍ ﾓﾄﾞﾙ'
32DD A09020AD
32E1 20B3A4BE
32E5 B9
32E6 0D                       DEFB   0DH
32E7                ;
32E7                          END
```



ラベル・テーブル

```
1BYTWR  31B0  ALLBYT  2F02  DATABF  4000  DBLOOP  313C  DBSTRT  3128
DTBLCK  310F  DTLOOP  31CD  ENDFG   3126  HBLKRD  301B  HDADDR  2F00
LSTBYT  2F04  MSG0    320D  MSG1    322B  MSG2    324D  MSG3    3258
MSG4    3266  MSG5    3288  MSG6    32A9  MSG7    32C6  MSG8    32D9
MSTRT   3000  PCWEND  3172  PLYCH   30BA  RDEROR  306B  RDLOOP  30D2
RDSTRT  300C  TLOOP0  30D4  TLOOP1  31B6  TLOOP2  31D5  TLOOP3  31E1
TLOOP4  31FA  WRINIS  307F  WRSTRT  309C
```

PC-8001



SAVEミス救済法



世田谷マイコンクラブ　**昭和一桁**

PC-8001では，SAVE時にうっかりCLOADと入力ミスしたり，暴走時R/Sをかけると，プログラムが消えてしまうことになっていますが，一般にプログラムがクリアされた状態は，エンド・マークや，ワーキング・メモリのデータのみが変化するだけのケースが多く，PCの場合もこの例に漏れず，次のような手順によって再生できます．

PRG・ENDを探す

プログラムは，メモリが16KBの場合，C020番地から**図1**のように入っています．

したがって，原理的には順を追っていけば，PRG・ENDを見付けることは容易ですし，ラインのENDマークが"00"になっているので，"00"附近を探していけばより容易です．

真のENDか

ENDアドレスが見付かったら，真のものか確認します．それには，ライン・ナンバーを確認するのが確実です．

たとえば，**図2**のような場合があります．

プログラムの最終ラインが780だったとすると，一見，C5BCからが最終ラインのように見えますが，ライン・ナンバーの06B8を逆算してみると，

（6×256＋B×16＋8）＝1720

となり，C5BCからは，いわゆるゴミであることがわかります．そこで，その前のC57Eを調べてみると，

（3×256＋12）＝780

で，ズバリ一致し，C57EからC5BAまでが，最終ラインに当たることがわかりましたので，C5BCから2～3個00に書き換えておきます．

ワーキング・メモリの修正

これからが苦心して探し出した，プログラム関係のワーキング・メモリで，**図3**のとおり修正してください．

以上で，ゴミの山になりかけた労作も見事再生できるはずですが，修正が不完全のままエディットやRUNさせると，完全にゴミの山となりかねないので，修正が終わったら，まず，CSAVEしておきましょう．

それと，大きいプログラムを入れる前に，1，2行のサンプル・プログラムを入れ，MONによってメモリ関係を良く調べておくことをお勧めします．

なお，本件に関し，＋αのデータなどあれば，ぜひ，編集部までお寄せください．

図1　C020からの内容

```
C020  00
C021  45C0  次のL#(ライン・ナンバー)のアドレス
C023  0A00  L#でこの場合は，10行を示している
  :
```

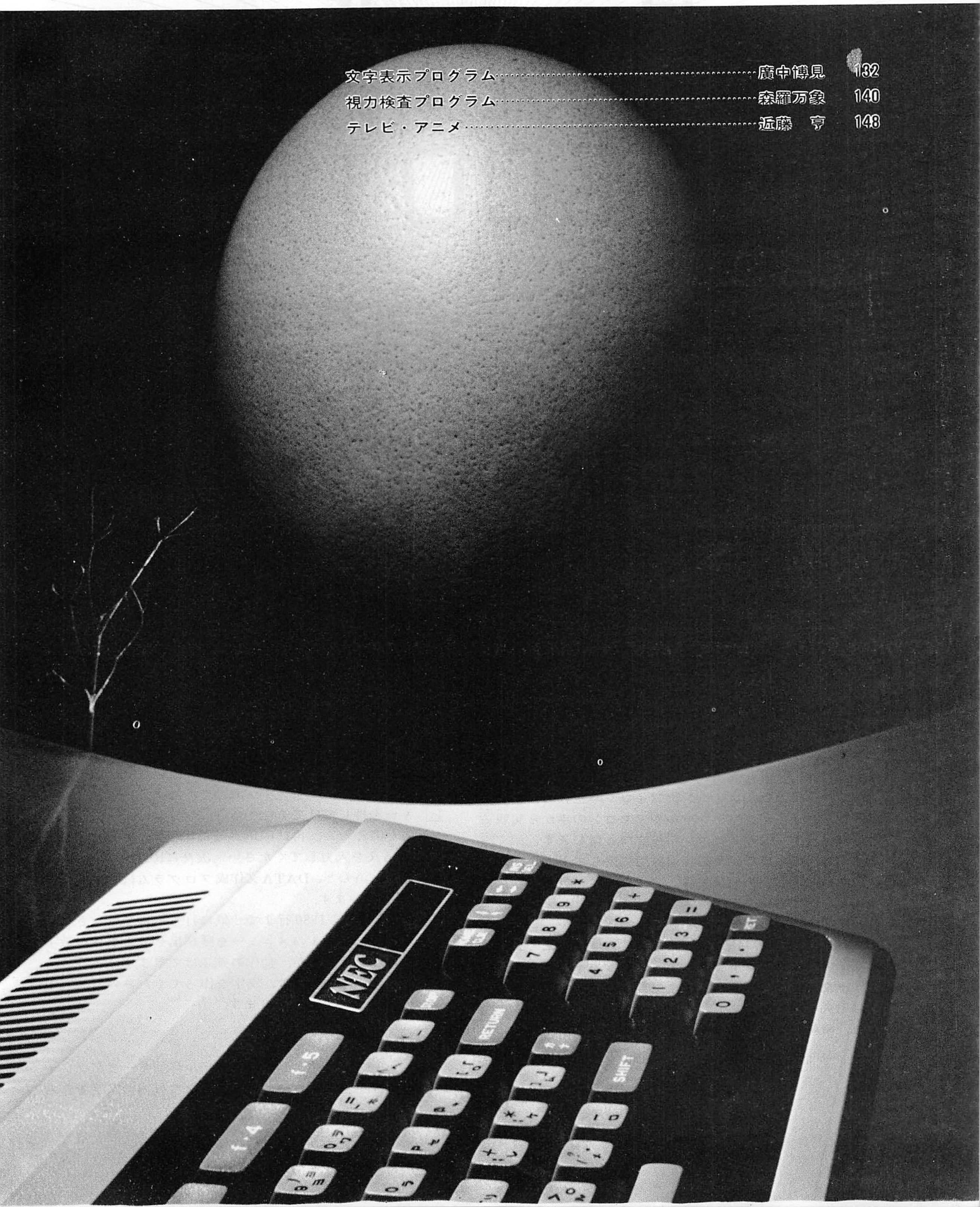
図3　ワーキング・メモリの修正

```
EFA0 ┐ プログラムの最終ラインのアドレス
EFA2 │ (先の例では，7EC5)
EFA4 ┘
EFA6  21C0
```

図2　真のENDかを確認する

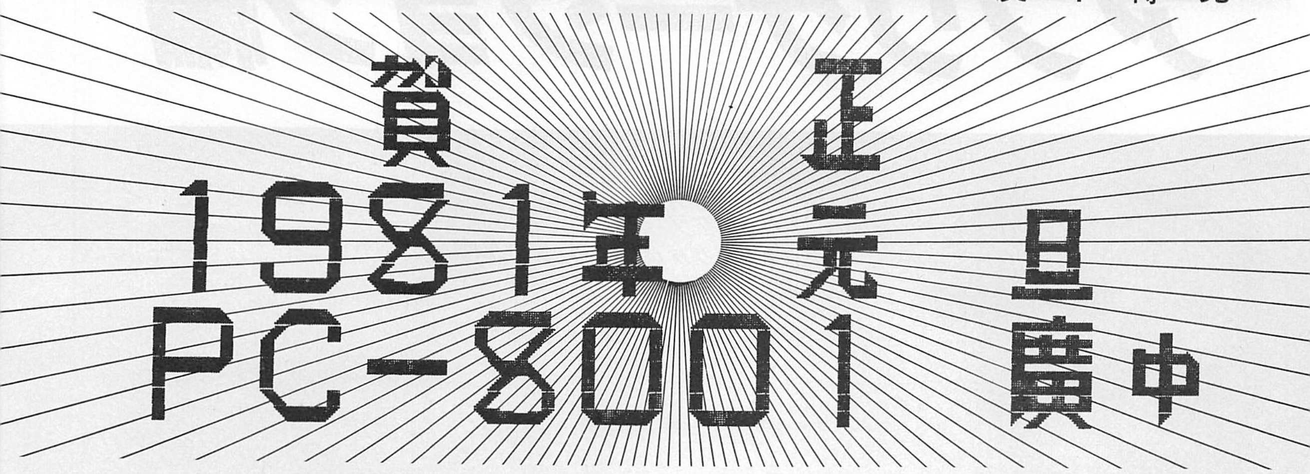
```
C546  007EC5F802
  :      ↓    L#76
C57D  00BCC50C03
        C57E  L#780
  :
C5BB  00DEC5B806
       0000に  L#1720
```

アプリケーション編



英数字
カタカナ
ひらがな
漢字
文字表示プログラム

廣中博見



『マイコンを買って何に使うの？』この意地悪な，しかも極めて的を得た鋭い質問に対して，何と答えたらよいでしょうか．世のナイコン亭主族は，将来マイコン・ウイドウになるかも知れない女房殿が，すんなりと納得して財布のひもをゆるめる解答を待ち望んでいます．

私の場合，『マイコンを子供の学習教材に使う』といった大義名分を立てて，'80年4月にPC-8001を購入しましたが，雑誌や市販のプログラム集はゲームばかりです．

そこで，長男にＡＢＣを教えるべく，ＡＢＣの大文字を作り始めたのが5月の連休のことでした．１文字ずつ形を作っては，家族の意見をきき修正する作業を繰り返し，ひらがなの『あいうえお』まで約100個の文字ができ上がったのが6月初旬でした．

さっそく，ＰＣを購入した『フルムラエルコン』へ行き，デモ用に展示してもらいつつ協力してプログラムを改良すること数回に及び，6月末には一応の完成をみました．

この時点で投稿しようと思ったのですが，しばらく使って使いやすいように直して発表するべく，試用期間をおきました．使っているうちに次々とアイデアが浮かび，以後2回の大改造で，収録文字数は410文字と増え，小学校1年生から3年生半ばまでの漢字を持つことができるようになりました．

このプログラムは，ナイコンorマイコン亭主族のために，子供の教育権をＴＶコマーシャルやママゴンの手から父親の手に取り戻す手段として大いに役立つと信じます．

2才から小学生まで，その発達段階に応じて1，2，3，ＡＢＣからWORD，あいうえお，アイウエオ，ひらがなによる文章，漢字の読み方，漢字まじり文の綴り方まで，総合的語学学習プログラムです．

プリンタ（TP-80/MP-80）を接続すれば，画面の文字文章をそのままプリントできます．大きなキャラクタ・グラフィック文字で，年賀状や暑中見舞のプリントができるように工夫されています．

また，時報付きのデジタル時計も内蔵されています．

プログラムの入力について

本プログラムには32K RAMが必要です．まず，800行から920行まで打ち込んでください．特にマシン語データとUSR関数の番地は決してミスがあってはなりません．よく確かめて，間違いがなかったらRUNさせてください．OKが出れば，

```
H=USR(0):?A$
```

をダイレクト・モードで実行してみてください．

画面上にASCIIナンバー32から247までの文字が印刷されれば正常です．ＯＫがでないときは，初めから入れ直した方が早いでしょう（したがって,RUNさせる前にテープに取った方が良いでしょう）．

OKが出たら，次にダイレクト・モードで，

```
H=USR1(0):FOR K=0 TO 21:?A1%(K),:NEXT
```

を実行してください．画面に0以外の数字が22個表われ，OKが出れば2つのUSR関数は正常です．

次に445行と446行を打ち込んでください．RUNさせると“DATAブンナンギョウメカラ？”と尋ねてくるので，**32**を入力してください．32行から54行までのDATA文が画面にプリントされ，カーソルが32行で点滅するので，［リターン］キーを押し続けてください．

入力が済んだら，再びRUNさせて**55**を入力して77行までDATA文を入力してください．次々にRUNさせて446行まで入力が済むと，DATA文作成プログラムは自動的に消滅してしまいます．

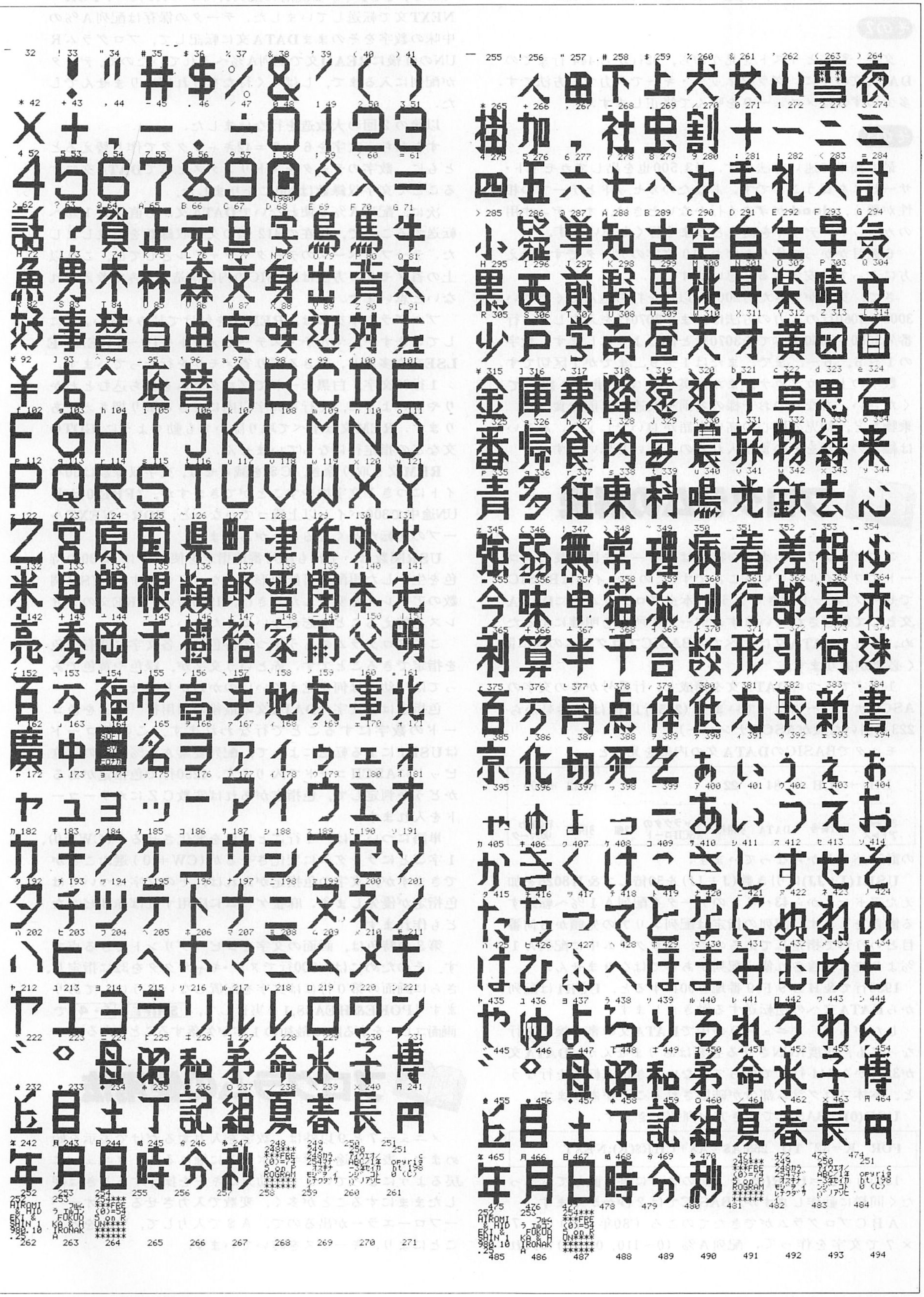
次に940行から1980行まで一気に打ち込んでください．この段階では1～4のメニューを呼び出しても小さな数字が表われるだけですが，動作の確認はできます．

さて，いよいよ難関です．文字を作ってください．文字の作り方に3つの方法があります．

その1

メニュー9を呼び出して，カーソル移動キーとグラフィック・キーを自由にあやつって，あなた好みの文字を作る．

キャラクタ例（DATA文に収録されている全キャラクタを印字）



時間はかかりますが，創造の楽しさ(？)があります．

その2

文字一覧表とリストを見ながら，32行から444行までのDATA文の中に，グラフィック・キーで入力する方法です．多少のずれはメニュー９を呼んで修正します．

その3

最も簡単で速い方法です．￥3,500也を出してカセット・サービスを買うことです．あなたのカセットとテープの相性が悪く，**cload**がうまく行かないときは，オーディオ用のカセット・デッキを使うとうまく行くと思います．

６箇月かかって作りあげた410文字のデータです．考え方によっては安すぎると思います．

さて，文字ができたら2000行以後を打ち込んでください．3000〜3060行の入力の方法は，まず3070行を入力して，行番号3000〜3060すべてを3070行と同じようにします．数字の１〜５，６〜␣まで，または１〜９␣までが１区切です．

数字の上にカタカナ，または英小文字で単語を綴ってください．あなたのお子様の名前や友達の名前や食べ物，乗物など，幼児の関心の高い単語が良いでしょう．あるいは絵本の文章をそのまま入れるのも良いと思います．

プログラムの解説

このプログラムの大きな特色は，データを固定長のレコードにファイルしていることです．このファイルはBASICでセーブ/ロードできるように，本文プログラム中にDATA文として保管されています．データの所在を明確にするため，これらDATA文はすべてのBASICプログラムの前に置く必要があります．

１文字で１つのDATA文を構成し，行番号がその文字のASCIIナンバーになっています（255行以後は行番号から223を引いた数がASCIIナンバー）．

モニタでBASICのDATA文の内部を見ると，

LH	+ LH	+ 84	+ 22 +	……	+ 22	+ 00
次の行のアドレス	行番号	DATA	引用符	キャラクタのASCIIコード　42個	引用符	行の終わりのマーク

の計50バイトからなっています．

USR1(J＋JJ)は引き数(J＋JJ)を50倍して＆H8026を加えたアドレスから43バイトのデータを配列A１％へ転送する働きをします．配列の指定は配列エリアの先頭から何番目という形で指定してあるので，プログラム中に配列A１％より先に宣言され使う配列があってはなりません．

1980行で＆HE９F９番地を00にすると，**USR1**は配列から**DATA**文への逆転送する働きをします．

したがって，メニュー９の中で**DATA**文の書き換えを行なっても，再度RUNさせる必要はありませんが，**DATA**文が32行から444行まで揃っていないときに逆転送を行なうと，本体プログラム部分が破壊されることがあります．

USR(0)はBASICで書き方が理解できます．

```
FOR K=32 TO 255:A$=A$+CHR$(K):NEXT
```

1315行の**H＝USR(0)**を上記のように書き直してもまったく同様に動作しますが，BASICでは２秒かかります．

ABCプログラムができたてのころ（'80年６月）は，７×７で文字を作って，配列A％（0〜110，0〜25）に入れておき，**PUT＠**する直前に配列A％からA1％へ，**FOR〜NEXT**文で転送していました．データの保存は配列A％の中味の数字をそのまま**DATA**文に転記して，プログラムRUNの直後に**READ**文で配列A％へ入れていたので，データが配列に入るまで，しばらく待たなければなりませんでした．

以後の２回の大改造を行ないました．

すなわち，文字を６×７＝42キャラクタで作り替えるとともに，数字のデータをストリングスとして**DATA**文にすることで文字収録数は200になりました．

次に，配列A％を使わないで**DATA**文から直接A１％へ転送することで，現在の412個の文字収録数を達成しました．テープ・ベースのデータ・ファイルとしては，これ以上の省メモリの方法はBASICの内部構造からみて考えられないと思います．

プログラム本文には，**REM**文を付けて読みやすいようにしていますが，マルチ・ステートメントや**IF〜THEN〜ELSE**文を多用し，できる限り省メモリを行なっています．

１行80文字，白黒モードでプログラムを打ち込むとわかりやすいように，１行を80字以内にできる限り押さえてあります．**REM**文はすべて取り除いても動くように，**GOTO**文などの指定行になっていません．

REM文を取り除き，C＄を減らせば，文字収録数は50バイトにつき１文字増やすことができますが，**?FRE(0)**がRUN途中で300バイト以上残っていないと，**FOR〜NEXT**ループの回転が遅くなることがあります．

USR関数はいずれも絶対番地指定を使わずに，Z80の特色を生かした再配置可能な形式になっていますが，**USR**関数のアドレスを変更したときは，1980行の**POKE**文のアドレスを替えることを忘れないでください．

このプログラムのもう１つの特色は，各文字に個有の色を指定できることです．赤という文字が，緑色や黄色であっては，幼児は何を覚えていいのかわかりません．

色指定は各文字の**DATA**文の最後の引用符『"』を色コードの数字にすることで行なわれます．この色コードは**USR1**による転送によって，配列要素A１％(21)の下位ビットにASCIIコードで入ります．1250行で色指定があるかどうか判定して，色指定があれば変数CZにカラーコードを入れます．

単語については，１行ごとに色を変化させるか(CW＝0)，１字ごとにランダムに変化させるか(CW≠０)選ぶことができますが，文字の色指定があれば，その文字については色指定が優先します．麻雀ゲームに応用すれば赤３ピンなども作れます．

第３の特色は，画面の文字をコピープリントできる点です．そのためには，900行でヌル・キャラクタを32に指定し，さらに画面の第０行には文字を配置しないようにしてあります（**POKE&HEA58,1**を実行して，**SHIFT** **f・4** で画面コピーをする際に最初の１字が欠落することがある）．

プログラムの使用法

メニュー７と９以外は，数字を入力するだけで表示を始めます．表示の途中でキーボードに触れると，メニューに戻るようになっています．幼児がキーを操作するときは押したままにすることが多く，変数で入力させると，オーバーフローエラーが出るので，A＄で入力して，VALをとることにより，キーミスを防いでいます．

写真 1　プログラムをRUNすると，まずこの表示が現われる．CR を押すと写真 4 へ．

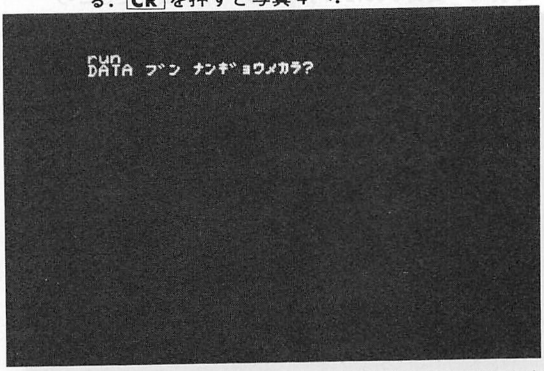


写真 2　写真 1 で32と入力すると，行番号32からDATA 文が現われる．

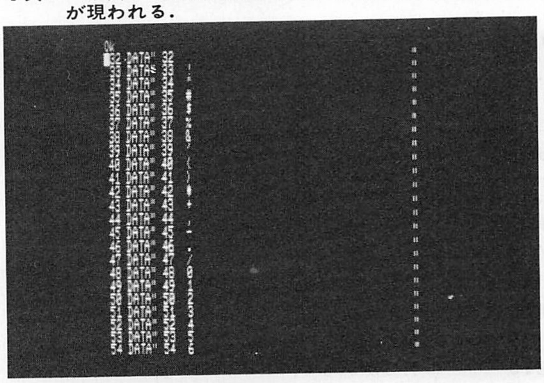


写真 3　文字パターンの入力

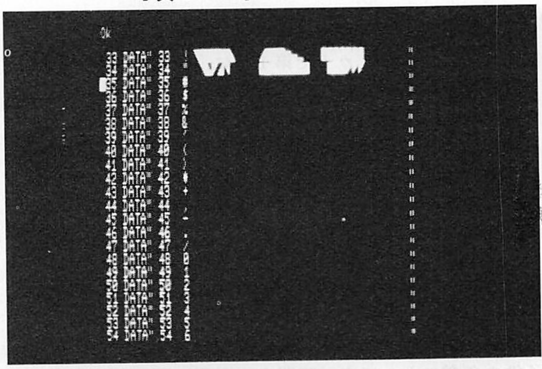


写真 4　メニューの表示．16種類のモードを選べる．

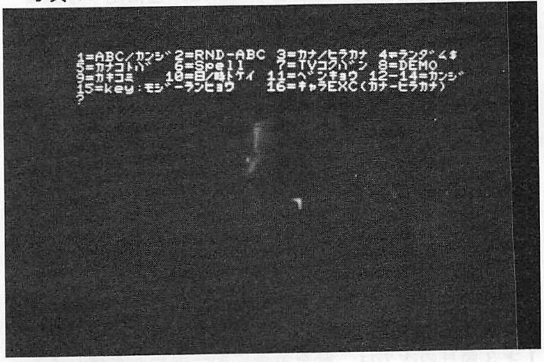


写真 5　メニュー 6 のSpellの実行例

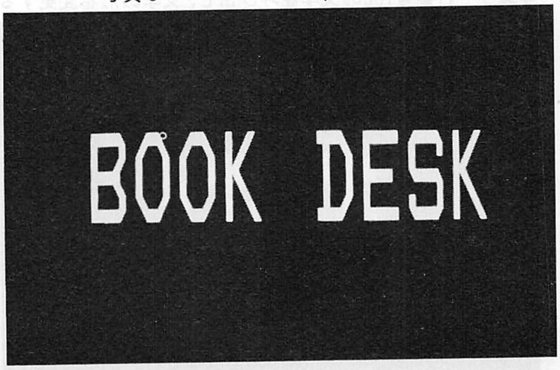


写真 6　メニュー10を呼び出す前の初期設定

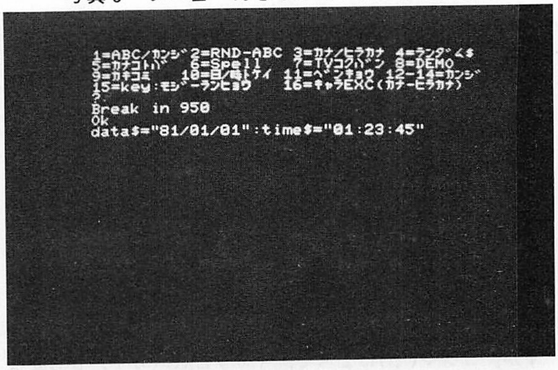


写真 7　メニュー10の実行例

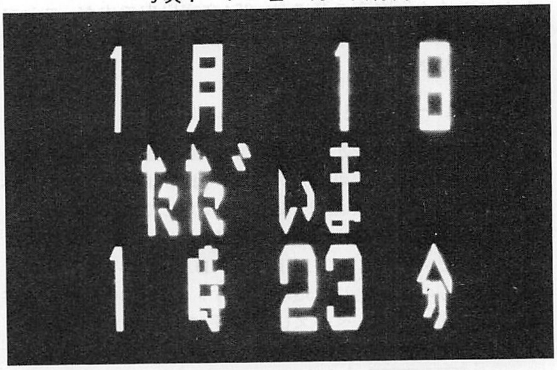


写真 8　メニュー15の実行例



写真９　行番号100の文字を呼び出し，文字の形を変更する．

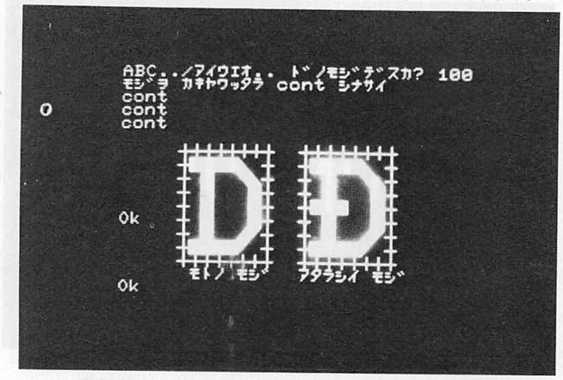
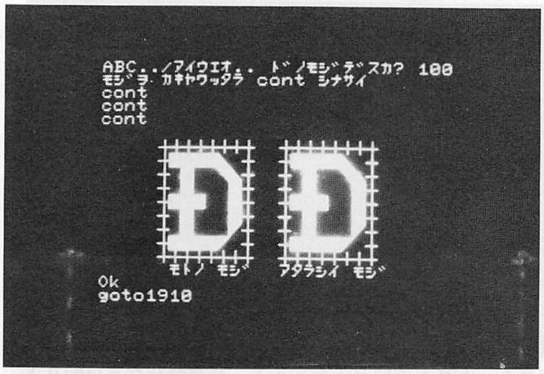


写真10　変更された文字を再び，呼び出したところ



●メニュー７　TVコクバンの使い方

プリンタを接続していない人は1710行の**LPRINT**の**L**を消してください．さもないと，プリンタがない場合やＳＷがOFFのとき，プログラムはここで立ち止まったままになります．

メニューの状態で，[7]を入力すると画面はクリアされ，80文字/行モードになり，カーソルがホーム・ポジションに点滅します．このとき，[リターン] キーだけを押すと，再びメニューに戻ります．何か文章を入力すると，その文章が30字区切りで画面に表示されます．

文字の種類はキーボードから入力できる文字に限られますが，コンマ(，)によって，キャラEXC（カナ，ひらがなモード切り替え）することにより，390 種の文字を使うことができます．文章の長さは，254文字以内ですが，コンマの数だけ少い文字数を入力しないと，途中でエラーが出ることがあります．

●メニュー９の使い方（写真９，10）

メニュー９に移ると，どの文字を書き換えるか，尋ねてきます．そこで，変更したい文字，またはその文字の行番号（ASCIIナンバまたは＋223)を入力してください．

[リターン]キーだけが入力されると，メニューに戻ります．入力された文字が２文字以上だと，**DATA** 文の行番号が入力されたものと判定して処理します．32以下の２桁の数字が入力されると，スペース(32)が呼び出されます．画面に現われた文字が呼び出したい文字と異なっていたときは，そのまま [リターン] キーを押します．

●メニュー15で文字一覧表をプリントする方法

１度メニュー７を呼び出して，プリンタの行間をつけるモードに設定した後，メニュー15を呼んでください．

文字が30字表示され，行番号とキーが表示されたら [ESC] で一時止めて，[SHIFT] [f･4] でプリントしてください．[SHIFT] [f･5] でライン・フィード行なうことができます．プリンドが終わったら [ESC] を押して解除して，次のプリントに移ってください．

７枚プリントが終わったら，メニューに戻って，キャラEXC（メニュー16）を実行して，もう１度メニュー15を呼んで，プリントを行なってください．アラビア数字が漢数字に，カタカナがひらがなになっているでしょう．

●メニュー10(時計，時報)の使い方

プログラムをRUNさせる前，または途中でSTOPしてダイレクト・モードで，

```
date$="80/01/03":time$="13:05:00"
```

のようにして，年月日時分秒を設定してください．

時間の表示は12時間表示です．不要な０は表示しません．正午は０時０分と表示されますが，時報は予鈴が短く，〝ピッピッピッ〟と３回鳴った後，12回BEEPします．

時計のメニューから抜け出すには，分も表示した後，２秒ぐらいいずれかのキーを押し続けると，2030行の**IF INKEY$=""** に引っかかって，コメントを表示した後，メニューに戻ります．

●メニュー11の使い方

英語の読み方をカタカナで第１行に，英単語を第２行に，第３行には日本語の意味をひらがなで表示します．

プログラム上では，途中にコンマを入れて，カタカナとひらがなのモード切り替えを行なっています．もちろん漢字とその読み方を表示するようにプログラムすることも容易です．Ａ＄の内容は各々，お好きな文章を入れてください．

変更法

子供の年齢によって表示時間や，画面に現われる文字の数の上限を変える必要があります．また，色指定の方法など主な変更箇所を**表１**に示します．その他の変更については，**表２**の変数表を参考にしてください．

おわりに

RFモジュレータの出力を３Ｃ２Ｖ同軸ケーブルで，カラーTVのUHF チャンネルに接続して，隣の部室から操作しています．２歳10箇月になる長男が，ＡＢＣの表示が終わってメニューが出てくると，『いち，りた～ん』と言いながら，自分でキーボードに入力するため，私のマイコン・ルームに入ってきます．

１才３箇月の次男は，無茶苦茶にキーボードを押しますが，ときにはプログラムが動き始めることがあります．する

表1　主な変更点

行番号	変　更　の　方　法
445～446	DATA文作成後は消除すること．
1020, 1040	AJ＝0 TO 50 文字の表示繰り返し数
1200	メニュー1～4, 12～14で表示する文字の数を制限する． FOR X＝15 TO 15＋RND(1)×8→1～8個の文字 FOR X＝15 TO 35 →15個で画面を埋める．
1210	FOR K＝0 TO 1000 メニュー1～4, 12～14の表示時間
1250	文字個有の色指定をしない場合には，IF文以後を消除すると，スピード・アップする．
1340	INKEY$＝" " でどのキーでもメニューへ． INKEY$<>"␣" で［スペース］キーでメニューへ．
1620	IF J＝14 THEN 1660→ピリオド以後を読み飛ばして，次のC＄の表示へ移る．IF文を削除すると，すべての文字を表示．
1640	メニュー5と6での表示時間設定ループ
1710	プリンタを接続しないときは，POKE文とLPRINT文を削除する．TMで表示時間を設定
1740	GOSUB 1240→音なし表示 GOSUB 1230→1文字ごとに音出し

表2　変数名表

変数	行番号	用　途
K	各所	FOR～NEXTループ・カウンタ，時間待ち
A$	各所	表示する文章，キーインされる文字，その他
JJ	1160，他	文字グループ交換用変数．JJ＝0でノーマル，JJ＝223で，ひらがな漢字モード
L	1200	文字を置く行を示す変数(L＝0,1,2)
C	1200	行のC番目に文字を置く．0～4または0～9
X	1200	文字を置く回数を示すループ変数
〃	1320，他	文章から文字を切り出す順番のループ変数
CX	1260，他	CX＝0で前回と同色，≠0で乱数で色決め．
CY	1260，他	色指定のとき0．乱数を使うときは7．
CZ	1260，他	色指定時カラーコード．その外のときCZ＝1
CW	1740，他	CW＝0で1行ごと，≠0で1字ごとに色変化する．
AJ	1020，他	ループ・カウンタ．表字繰り返し回数
N	1360，他	文字一覧表のページNを示す．N＝X/27 −1
JK	1510 ～1640	JK＝5→1行5文字表示．Width40モード JK＝10→1行10文字表示．Width80モード
〃	1900 ～2000	メニュー9で文字を呼び出すとき，行番号で文字指定する際の補正値．0または223.
TM	1750，他	表示時間をコントロールする変数
KM	1360	1ページに30タイトル印字するためのカウンタ
M	1360	1行に10タイトル印字するためのカウンタ
A	950	A＄で入力した文字のVAL
JM	1240	JM＝0でノーマル，JM＝−223でJM＋JJ＝0になる．

と『でけた』と言って手をたたいて喜こびます．ただし，**STOP** キーも時々押すので，仕事から帰ってみると，0～10行に変な文字が入力されていることもあります．

長男は1箇月ほどで，ABCが読めるようになり，3箇月でひらがなが半分ほど読めるようになりました．表示時間を発達段階に応じて，こまめに調整し，画面の文字が何であるかを一緒に声を出して教えることが必要です．

文字の覚え方は，りんごの『**り**』，海苔の『**の**』，船の『**ふ**』と言ったように具体的な物と結びつけて記憶しているようです．また，パパの『**H**』，ママの『**Y**』，しんくんの『**S**』などと頭文字を結びつけてABCも覚えました．

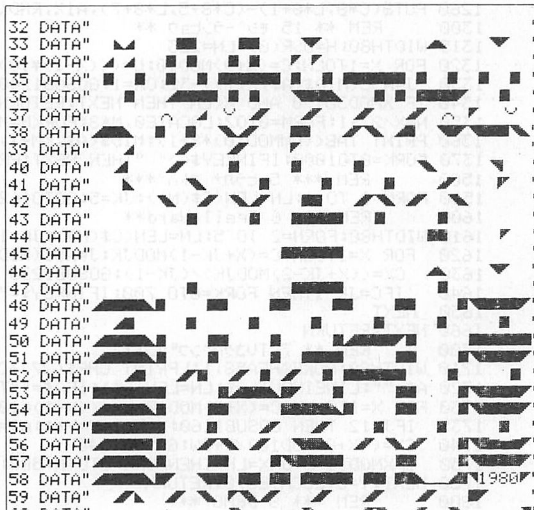
近ごろでは，女房殿も『高精度カラーモニタを買っても良かったね』などと申すようになりました．

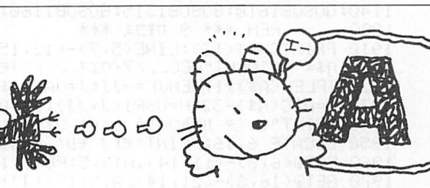
私のシステムはPC-8001(32K)とグリーン・モニタとOKIタイパー5200です．近いうちにPCG-8100を買いたいと思っています．そして，次にミニ・フロッピーを考えています．しかし，誰もが簡単に使えるシステムというのは，ここ1年はテープ・ベースのBASICだと思います．

このプログラムの他に，住宅管理組合（118戸）管理プログラム，病理標本管理プログラムなどを作成していますが，いずれも本プログラムの**USR1**と同様に**DATA**文から直接に配列や変数に転送することで，取り扱うデータ量を増すように改造するつもりです．

また，本プログラムを外国語学習用として，アラビア文字，ギリシャ文字，ヒンディ文字（サンスクリット），ハングルなどのキャラクタを作って活用したいと思っています．

データ文の一例





◘参考にしたもの

1)　寺田浩詔監訳：Z-80マイクロコンピューター，丸善

◘プログラム・リストの変更について

1)行番号920に次の文を追加してください．

```
DIM A2%(6):FOR K=0 TO 6
:A2%(K)=8224:NEXT
```

2)次の2行のように変更すると，文字10個目にベース・カラーが出なくなります．

```
1260 CC=RND(CX)*CY+CZ
1270 PUT@(C*8,L*8+1)-(C*8+5,L*8+7),A2%,CC:PUT@(1*8+6,L*8+1)-(1*8+7,L*8+7),A2%,CC:RETURN
```

3)MP-80 TYPE2を使われる方は，行番号1710のLPRINT以降を次のように変えてください．

```
CHR$(27)+"1"+chr$(27)+"A"
+chr$(7)+chr$(15):TM=15
```

（chr$(15)：縮少文字）

以上のように変えないと，MP-80のときにグラフィック文字の間があいてしまいます．



文字表示プログラム・リスト

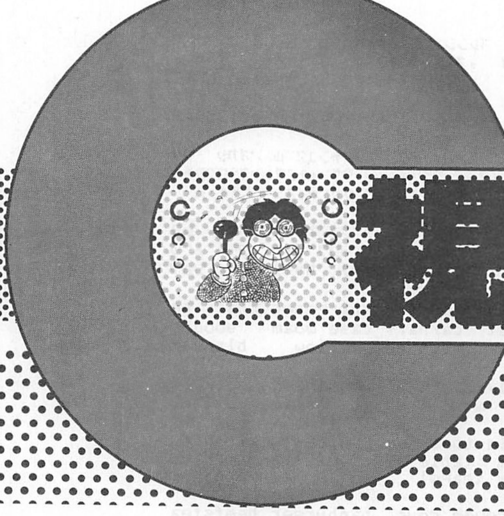
```
445 INPUT"DATA ﾌﾞﾝ ﾅﾝｷﾞｮｳﾒｶﾗ";JJ:IFJJ=0THEN800 ELSE IFJJ<32 THENJJ=32
446 WIDTH80:FORJ=1TO23:LOCATE0,J:J$=STR$(J+JJ-1):PRINTJ$;TAB(4);"DATA"+CHR$(34);
J$,CHR$((J+JJ+190)MOD223+32);TAB(51);CHR$(34):NEXT:LOCATE0,0:END
800 DATA2a,a0,ef,01,08,00,09,3e,df,47,77,23,3e,A0,77,23
810 DATA3e,e8,77,21,A0,E8,3e,20,77,23,3c,05,20,FA,c9,11
820 DATA26,80,2A,A8,F0,29,EB,19,EB,29,29,29,EB,19,EB,29
830 DATA19,EB,2A,A2,EF,01,07,00,09,EB,01,2C,00,ED,B0,c9
900 CONSOLE0,25,0,1:WIDTH40,25:COLOR4,32,0:REM   **** MAIN  ROUTIN  ****
910 CLEAR450,&HE9BF:DIMA1%(21),C$(5):DEFUSR=&HE9C0:DEFUSR1=&HE9DF:DEFINT A-Z
920 RESTORE800:FORK=0TO63:READA$:POKE&HE9C0+K,VAL("&H"+A$):NEXT:KEY1,"width80:"
940 GOSUB3000:KEY2,"console,,,0"+CHR$(13):CY=7:CZ=1
950 WIDTH40:INPUT"1=ABC/ｶﾝｼﾞ2=RND-ABC 3=ｶﾅ/ﾋﾗｶﾅ 4=ﾗﾝﾀﾞﾑ$ 5=ｶﾅｺﾄﾊﾞ  6=Spell   7=T
Vｺｳｶﾊﾞﾝ 8=DEMO   9=ｶｷｺﾐ   10=日/時ﾄｹｲ 11=ﾍﾞﾝｷｮｳ 12-14=ｶﾝｼﾞ15=key:ﾓｼﾞｰﾗﾝﾋｮｳ   16=ｷｬﾗ
EXC(ｶﾅｰﾋﾗｶﾅ)  ";A$:A=VAL(LEFT$(A$,2)):PRINTCHR$(12)
960 ON A GOSUB 1010,1020,1030,1040,1510,1610,1710,1810,1910,2010,2110,1120,1130,
1140,1315,1160:GOTO950
1000     REM ** 1-4,12-14 **
1010 FOR J=64TO 90:GOTO1200
1020 FOR AJ=0TO 50:J=RND(1)*26+65:GOTO1200
1030 FORJ=145TO191:GOTO1200
1040 FOR AJ=0TO 50:J=RND(1)*53+145:GOTO1200
1120 FOR J=0 TO 64:GOTO1200
1130 FOR J=91TO133:GOTO1200
1140 FORJ=192TO223:GOTO1200
1160 IF JJ=0 THEN JJ=223:RETURN ELSE JJ=0:RETURN
1200 CX=RND(1)*3+1:FORX=15TO15+RND(1)*8:L=(X+J+CX)MOD3:C=((X+J)¥CX)MOD5:GOSUB123
0:NEXT
1210 FOR K=0TO1000:IFINKEY$=""THEN NEXT:PRINTCHR$(12):NEXT:RETURN ELSE RETURN
1220      REM **  PUT@  **
1230 BEEP1:FORK=0TO5:NEXT:BEEP0
1240 IFJJ+J>412THEN JM=-223 ELSE JM=0
1250 H=USR1(J+JJ+JM):IFA1%(21)<>34THENCY=0:CZ=(A1%(21)-28)MOD7+1:ELSE CY=7:CZ=1
1260 PUT@(C*8,L*8+1)-(C*8+5,L*8+7),A1%,RND(CX)*CY+CZ:RETURN
1300     REM ** 15 ﾓｼﾞｰﾗﾝﾋｮｳ **
1315 WIDTH80:H=USR(0):LN=223
1320 FOR X=1TOLN:C=(X+9)MOD10:L=(((X-1)¥10)+3)MOD3:J=ASC(MID$(A$,X,1))-32
1330  J=ASC(MID$(A$,X,1))-32:CX=1:GOSUB1240
1340 IF XMOD30<>0 AND X<LN THEN NEXT:RETURN
1350 N=X/27-1:FORM=0TO2:LOCATE0,M*8:FORKM=M*10+30*N TO9+M*10+30*N
1360 PRINT TAB((KMMOD10)*8+1);MID$(A$,KM+1,1);KM+32+JJ;:NEXT:NEXT
1370 FORK=0TO1000:IFINKEY$<>" "THEN NEXT:PRINTCHR$(12):NEXT:RETURN ELSE RETURN
1500      REM *** 5 ﾋﾗｶﾅ ｺﾄﾊﾞ***
1510 FORN=0 TO 1:LN=LEN(C$(N)):JK=5:GOTO1620
1600      REM *** 6 Spell/Ward**
1610 WIDTH80:FORN=2 TO 5:LN=LEN(C$(N)):JK=10:IFLN=0 THEN RETURN
1620  FOR X=1TOLN:C=(X+JK-1)MODJK:J=ASC(MID$(C$(N),X,1))-32:L=1:IFJ=14 THEN1660
1630   CX=((X+JK-2)MODJK)/(JK-1):GOSUB1230
1640   IFC=JK-1THEN FORK=0TO 700:IFINKEY$=""THENNEXTK:PRINTCHR$(12) ELSE RETURN
1650  NEXT
1660 NEXT:RETURN
1700      REM ** 7 TVｺｳｶﾊﾞﾝﾌﾟﾘﾝﾄ*
1710 WIDTH80:POKE&HEA58,1:LPRINT CHR$(27)+CHR$(49):TM=15
1720 A$="":LINEINPUT A$:LN=LEN(A$):IFA$=""THEN RETURN ELSE GOSUB1730:GOTO1720
1730 FOR X=1TO LN:C=(X+9)MOD10:L=(((X-1)¥10)+3)MOD3:J=ASC(MID$(A$,X+Y,1))-32
1735  IFJ=12 THEN GOSUB1160:X=X-1:Y=Y+1:A$=A$+" ":NEXT
1740  CX=((X+8)MOD10)/9+CW:GOSUB1240
1750  IFXMOD30=0 ORX=LN THENFORK=0TOTM*500:NEXT:IFLN<>30 THEN PRINTCHR$(12)
1760 NEXT:Y=0:LOCATE0,0:RETURN
1800      REM *** 8 DEMO ***
1810 GOSUB1010:GOSUB1020:GOSUB1030:GOSUB1040:GOSUB1510:GOSUB1120:GOSUB1130:GOSUB
1140:GOSUB1610:GOSUB1315:GOSUB1160:GOSUB1315:JJ=0:GOSUB2110:GOSUB2010:RETURN
1900      REM *** 9 ｶｷｺﾐ ***
1910 PRINTCHR$(12):LINE(5,7)-(12,15),"+",B:LINE(15,7)-(22,15),"+",B
1920 A$="":INPUT"ABC../ｱｲｳｴｵ..  ﾄﾞﾉﾓｼﾞﾃﾞｽｶ";A$:IFA$=""THEN RETURN
1925 IFLEN(A$)>1THENJK=-JJ:J=VAL(A$)-32:IFJ<0 THENA$=" "ELSE H=USR1(J):GOTO1940
1930 J=ASC(A$)-32:H=USR1(J+JJ):JK=0
1940 PRINT"ﾓｼﾞｦ ｶｷﾔﾜｯﾀﾗ cont ｼﾅｻｲ":PRINT"cont":PRINT"cont":PRINT"cont"
1950 LOCATE 6,16:PRINT"ﾓﾄﾉ ﾓｼﾞ  ｱﾀﾗｼｲ ﾓｼﾞ":PRINT:PRINT"goto1910":LOCATE0,17
1960 PUT@(6,8)-(11,14),A1%,5:PUT@(16,8)-(21,14),A1%,7:END
1970 GET@(16,8)-(21,14),A1%:IFJ+JJ+JK>412 THEN950
1980 POKE&HE9F9,0:H=USR1(J+JJ+JK):POKE&HE9F9,&HEB:GOTO1910
2000      REM *** 10 ﾄｹｲ,時ﾎｳ ***
2010 WIDTH80:PRINTCHR$(12):JJ=0
2020 A$=RIGHT$(STR$(VAL(MID$(DATE$,4,2))),2)+" 月 "+RIGHT$(STR$(VAL(MID$(DATE$,7,
2))),2)+" 日  , ﾀﾀﾞｲﾏ  ,  "+RIGHT$(STR$((VAL(LEFT$(TIME$,2)))MOD12),2)+" 時 "+RIGHT
$(STR$(VAL(MID$(TIME$,4,2))),2)+" 分 ":LN=30:CW=1:TM=1:GOSUB1730
2030 Z$=MID$(TIME$,4,2):IFZ$="59"THEN2050 ELSE IFZ$="30"THEN2040 ELSE IF INKEY$=
""THEN2020 ELSE 2060
2040 IF VAL(RIGHT$(TIME$,2))>2  THEN2020 ELSE BEEP:PRINTCHR$(12):GOTO2020
2050 IF VAL(RIGHT$(TIME$,2))<55 THEN2020 ELSET$=TIME$:IFVAL(RIGHT$(TIME$,2))<57
THEN2050 ELSE FORI=0 TO 3 :BEEP1:FORX=0 TO 5:NEXT:BEEP0:FORX=0 TO800:NEXT:NEXT:F
ORX=1 TO(VAL(LEFT$(T$,2)))MOD12+1:BEEP:FORI=0TO800:NEXT:NEXT:PRINTCHR$(12):GOTO2
020
2060 BEEP:A$=" pc-8001   n-bacic  :ｱｲｳｴｵ､｢｣  ,ｺﾚﾗﾉ,MD,ﾊ,    UV,ﾆｮｷ,T,ｴﾙ,   S,ｶﾞ,O,
```



```
oﾏｽ,.  ":LN=60:TM=15:GOSUB1730:RETURN
2100     REM  11 ｵﾍﾞﾝｷｮｳ ***
2110 WIDTH80:A$="  ｱｯﾌﾟﾙ     ,apple    , ﾘ ﾝ ｺﾞ,      ｵﾚﾝｼﾞ    orange    , ﾐ ｶ ﾝ
,  ｽｸｰﾙ     school     ???     ｱﾐ  ﾌﾚﾝﾄﾞ ami friend  ,ｵ,ｲ,ﾀﾞﾁ,      ｱｲ ｵｰ   i'o (
I,T-,)  ﾏｲｺﾝ,ﾉ, ■     ｻﾝ  ﾑｰﾝ  sun moon , ﾀｲﾖｳ,   月      ":LN=180:TM=3:CW=0:GOSUB17
30
2120 WIDTH80:A$="  @   A   1981年 B C pc-8001 「」     ":LN=30:TM=9:CW=0:GOSUB1730:R
ETURN
3000 C$(0)="ｼﾝｲﾁﾛﾋﾃﾞｷ ﾋﾛﾐ  ﾕｳｺ  ｷｮｳｺ  ﾄﾓｴ  ﾀｶｼ  ﾅｵﾐ  ｶﾅｺ ﾘｮｳｺ ﾐﾅﾁｬﾝﾕｽﾞﾙ  ｵｶｼ  ｳｻ
ｷﾞﾄﾗｻﾝ ﾛﾊﾞｻﾝﾏ ﾏ  ｺﾛｺﾛ ﾌﾄｯﾀ ﾊﾟﾝﾀﾞｺﾛﾝﾃﾞｱｿﾌﾞﾉﾀﾞｲｽｷｱﾌﾞﾅｲｱ ｲ ｽｸﾘｲﾑ ｿﾞｳｻﾝｲﾁﾆｻﾝｼｺﾞﾛｸｼﾁﾊ
ﾁｸｼﾞｭｳ ｷ ｲ ﾛｵﾚﾝｼﾞﾓﾓｲﾛ ｱｶｱｵｷﾜﾝﾜﾝ ﾃﾞﾝｼｬﾋｺｳｷ ﾄﾞｳﾛ ｼﾞｭｳｽｼﾝｲﾁﾛｷﾘﾝ  ｺﾞﾘﾗ ﾗｲｵﾝ ﾊﾞﾅﾅ ﾊﾞ
ﾊﾞﾐｷﾞﾃ ﾋﾀﾞﾘﾃ
3010 C$(1)="ｶｲﾓﾉ ｸﾀﾞｻｲｾﾝｴﾝ ﾌｱﾌｱ ｵﾌﾄﾝ ﾈﾝﾈｽﾙｵｼﾘ  ｼﾞﾛｳ ﾁｮｳﾁｮ ﾑｼ  ﾀﾞﾝｺﾞﾎｯﾍﾟ ｻﾝｶｸ  ｼｶ
ｸ ﾏ ﾙ  ﾂﾐｷ  ｶﾀﾂﾞｹｼｬｼﾝ ﾌﾞﾄﾞｳ ﾒﾛﾝ ｽｲｶ  ｽﾎﾞﾝ ﾎﾞｸ  ｺｳｴﾝ ｽﾍﾞﾙ ﾃﾂﾎﾞｳｶﾞｯｺｳ ｱｯﾁ  ｺｯﾁ ﾄﾞｯ
ﾁ ｺｺﾄﾞｺﾄﾞｺｲｸﾊﾟﾊﾟ ｼｺﾞﾄ ｻﾖﾅﾗ ﾏﾀﾈ  ｵﾊﾖｳ ｺﾝﾆﾁﾜﾉﾘﾓﾉ ﾊﾞｽ  ｶﾗｽ  ﾅｾﾞ  ﾅｸﾉ  ｶﾗｽﾉ ｶｯﾃﾃﾞｼｮｳ
 ｶﾜｲｲ  ﾅﾅﾂﾉｺ
3020 C$(2)=" sun moon  building  book desk bookstand  pencil  sharpener chair be
d  carpet   curtain  rooster   chick    bird nest  carnary cage woam   good
  morning   mirror   wall door parrot    wake up    picture  pillow    blanket
  pajamas ?
3030 C$(3)="ｲﾛﾊﾆﾎﾍﾄﾁﾘﾇﾙｦﾜｶﾖﾀﾚｿﾂﾈﾅﾗﾑｳｲﾉｵｸﾔﾏｹﾌｺｴﾃｱｻｷﾕﾒﾐｼｴﾋﾓｾｽﾝ「」,goodnight baby ok
?  swing   slide downbanana    apple    orange   muscat of  hiroshima  3+3=6
  wine sake summer finsun moon  star   end
3040 C$(4)="plate tarofrying pan mother   pot kettle tooaster  juicer   kitchin
  knife cut big sistertea milk  egg bread juice jam  table    cheese in sugar cu
t salad dishbaby dog  fork spoonknife cup lunch box teapot    glass of  water ho
t spaghetti?
3050 C$(5)="sandwich  cup of teaomelet onecurry riceveg soup  hanburger beefstea
k telephone sky  bird   nurse   doctor    hospital   bank     cloud    policema
n bus driverschool bus  womam   overcoat  sidewalk   boots    car truck sportsca
r triangle.
3060 RETURN
3070 C$( )="123456789 123456789 123456789 123456789 123456789 123456789 12345678
9 123456789 123456789 123456789 123456789 123456789 123456789 123456789 12345678
9 123456789 123456789 123456789 123456789 123456789 123456789 123456789 12345678
9 123456789 "
```

PC-8001

森羅万象



視力検査プログラム

視力検査はいろいろな目的で行なわれます．たとえば，目の健康診断，各種の適性検査など広い分野で実施されています．本プログラムはこれをマイコンで行なおうとするものです．

1 視力検査について

日本の視力検査に関する規定は，以前には1909年の国際眼科学会における決定と，1938年の日本眼科医会における決定に準拠していました．

そして1964年，今日の実情に即した新しい視力検査基準が，文部省科学研究費総合研究視力研究班により設定されました．本プログラムも基本的にこの基準に従っています．

写真1　システム全景



図1　Landolt環視標

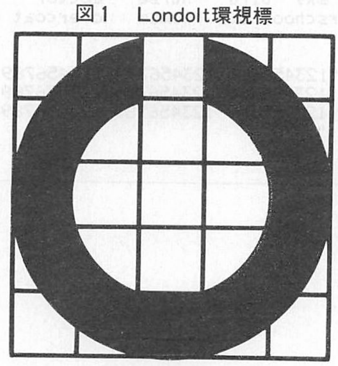


表1　視力検査基準表

準標準域値	
視標数	正答数
1	1
2	2
3	3
4	3
5	4

この基準[文献1]ではLandolt環を標準視標としています．このLandolt環とは図1に示すもので，環の外径7.5mm，太さおよび切れ目幅1.5mmのものを検査距離5mで用いたときの視力値を1.0とします．視力表には標準検査装置，準標準検査装置，特殊検査装置の3種類があります．

本プログラムでは実用性に重点を置き，視力の値を特にやかましく問題としない場合，たとえば眼鏡検定を主目的とする視力検査，あるいは集団検査，スクリーニング検査などに用いられる準標準検査装置の基準に従っています．

この場合，Landolt環の寸法の許容差は標準値の±10%以内で，本プログラムもほとんどこの許容差内です．

視標の段階のきざみ方は特に定められていませんが，グラフィックの1ドットが約1.5mm×1.5mmなので，細かく段階を刻むわけにもいかず，

0.1　0.15　0.2　0.25　0.3　0.5　1.0

の7段階としました．

検査結果の判定基準は**表1**に示すとおりで，本プログラムでは，

視標数：4　**正答数**：3

としました．すなわち，同じ視力の視標を4回表示して（もちろん環の切れ目の方向は毎回同じとは限らない），3回以上正答数が得られた最小視標の視力値をもって視力とするものです．

システム構成は，PC-8001とグリーン・ディスプレイP

写真２　検査のデモンストレーション(その１)

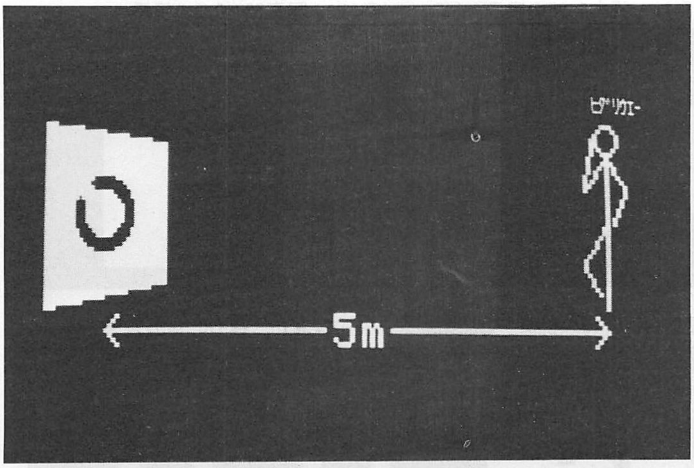
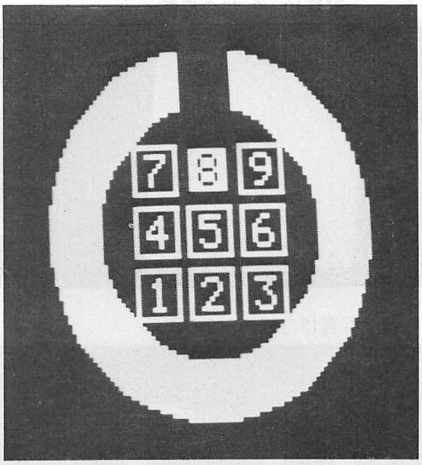
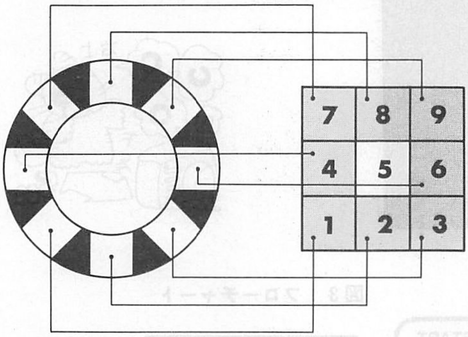


写真３　検査のデモンストレーション(その２)



C-8041です(**写真1**)．ディスプレイには必ず12インチの物を使用してください．他のインチの物を使用すると視標の大きさが違ってきます．なお，PC-8001とN-BASICに関しては，文献2)～5)を参考にしてください．

図２　切れ目の方向と対応するキー



2 使用方法

プログラムをRUNさせると**写真２**のデモ１と**写真３**のデモ２とを交互に繰り返します．このときに，RETURNキーまたは任意の英数キーを押すと，デモから抜け出し，検査が始まります．

視標は，小さいものから大きいものへと表示されます．視力で言えば，一番最初に1.0の視標が，そして一番最後には0.1の視標が表示されるわけです．表示される視標の切れ目の方向がわかれば，対応するキーを押します(**図２**)．わからなければ，5のキーを押します．

表１に示す正答数を満足すれば，そのときの視力が表示され，正答数を満足しなかったり，わからない場合（5のキーが押された場合）には，一段階大きい視標が表示されます(**写真４～写真17**)．

写真４　1.0の視標

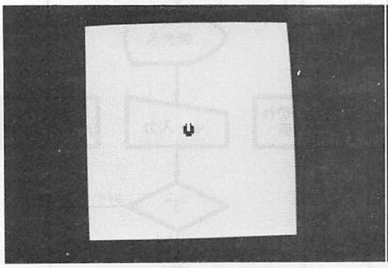


写真６　0.5の視標

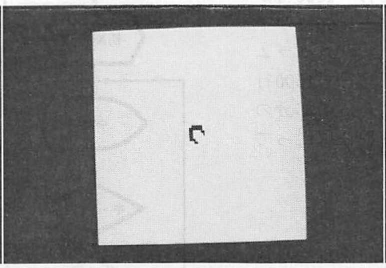


写真８　0.3の視標

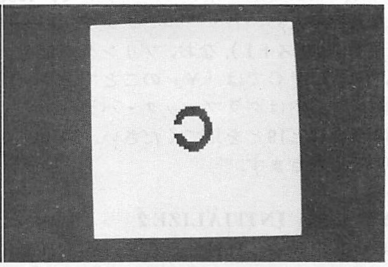


写真５　視力1.0の表示



写真７　視力0.5の表示



写真９　視力0.3の表示



写真10　0.25の視標

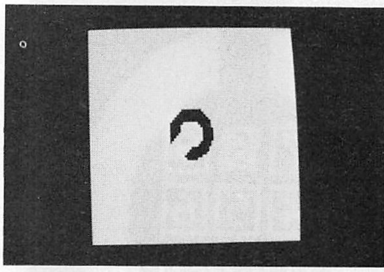


写真11　視力0.25の表示



写真12　0.2の視標

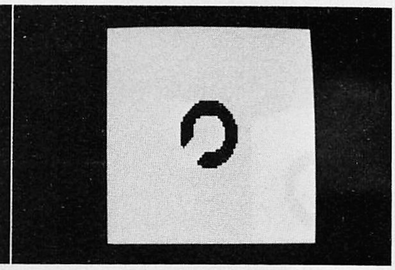


写真13　視力0.2の表示



写真14　0.15の視標

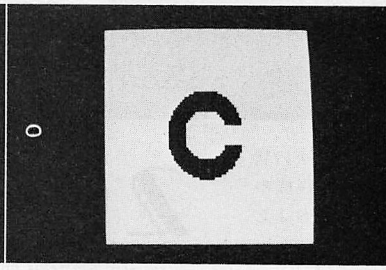


写真15　視力0.15の表示



写真16　0.1の視標

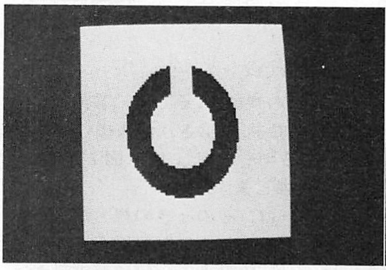
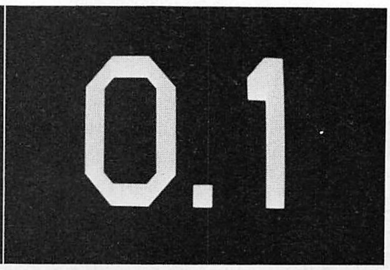


写真17　視力0.1の表示





3 プログラム

10行～1080行がメイン・ルーチンで，1090行～1500行が視力表示のための各サブルーチンで，1520行～1940行がデータです（**リスト1**）．なお，プリンタの都合上，プログラム中，『＼』はPCでは『￥』のことであり，1090行～1500行中の『　』内はグラフィック・パターンです．この部分のみは**写真18**と**19**とを見てください．以下，行番号に従って説明していきます．

①120～160　INITIALIZE 2

COLOR，CONSOLE，WIDTH，CLEAR，それに型宣言と配列宣言です．

②170～430　DATA READ

DATA文のREADと，PUT＠のために配列の一番最初（添字＝0）にドット数の設定を行ないます．**表2**に配列名とその内容とを示します．

ここで少し，PUT＠でグラフィックを表示させるためのデータの作成の方法について触れます．通常PUT＠は，GET＠で配列へ読み込んだキャラクタやドット・グラフィックスを画面の任意の位置へ表示させるのに使用されます．

したがって，GET＠で配列へ読み込まれるのと同じ規

図3　フローチャート

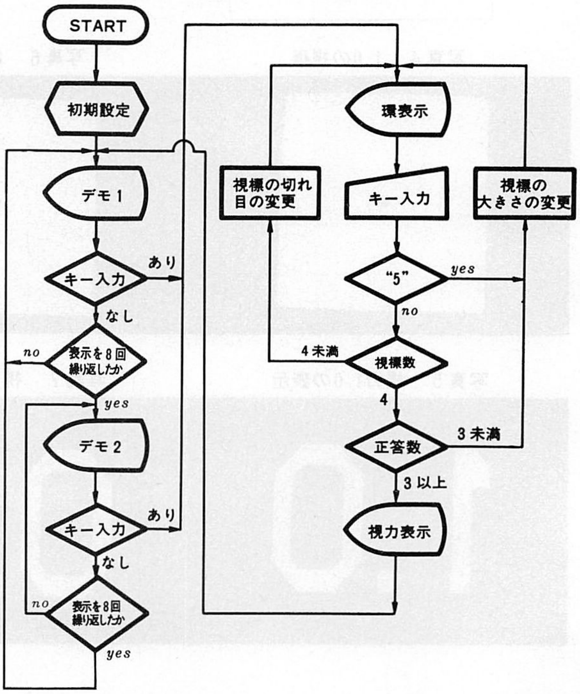


表2　配列名とその内容

配列名	内　　容
N 0	N 4，N 5，N 6，N 7をＰＵＴ＠で表示させる場合には，このＮ０に転送してからＰＵＴ＠で表示させる．
N I	デモＩの『人間』をＰＵＴ＠で表示させるためのデータ．
N 2	デモＩの『矢印』と『距離』とをPUT＠で表示させるためのデータ．
N 3	デモＩの『視標のバック』をPUT＠で表示させるためのデータ．
N 4	デモＩの『視標』をPUT＠で表示させるためのデータ．
N 5	デモ２の『数字キー』をPUT＠で表示させるためのデータ．
N 6	デモ２の『視標』をPUT＠で表示させるためのデータ．
N 7	デモ２の『切れ目』をPUT＠で表示させるためのデータ．
N 8	同　　上
N 9	同　　上
P	N６(０～3,i)のデータからN６(４～7,i)のデータを作成するときに使用するデータ．
Q	デモＩの『ウエー』，『ミギー』などのデータ．
S	視標がこの配列を使用して作成される．
L	デモ２でN０，N５～N９などをPUT＠するときの，パターンの左上の位置のデータ．
M	視標を作成するためのデータ．

則で（ＧＥＴ＠を使わずに通常の代入方法，たとえば『＝』やＲＥＡＤ文で），配列へグラフィック・データを代入させれば，それをＰＵＴ＠で画面の任意の位置へ表示させることができるわけです．では，具体的に説明しましょう．

図４は，デモ１に使用している『人間』を，パターン図からグラフィック・データに変換して配列Ｉ％に入れる場合です．

まず，パターン図を左上から右下へ図（ａ）の方向に16ドットずつ区切って行きます．そして一番最初，つまり左上の１ドットを含む16ドットをＩ％(1)に（注：Ｉ％(0)ではない），そして次をＩ％(2)にと順々に入れて行きます．

入れ方は同図（ｃ）に示すように，16ドットの左側がＬＳＢ，右側がＭＳＢになるように変換します．そして，配列の一番最初，つまりＩ％(0)にドットの総数を入れておきます．

写真18　行番号1090～1320のリスト

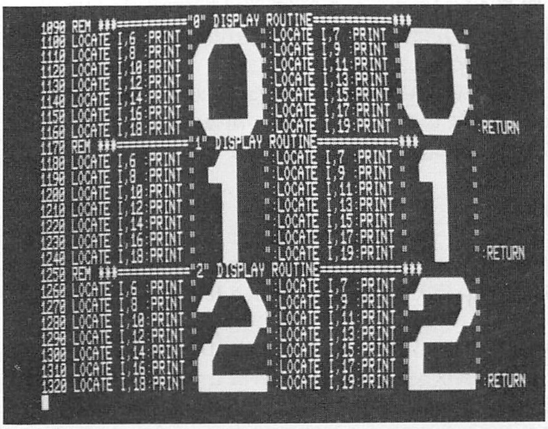
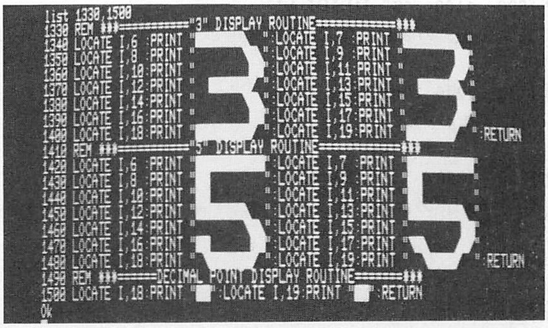


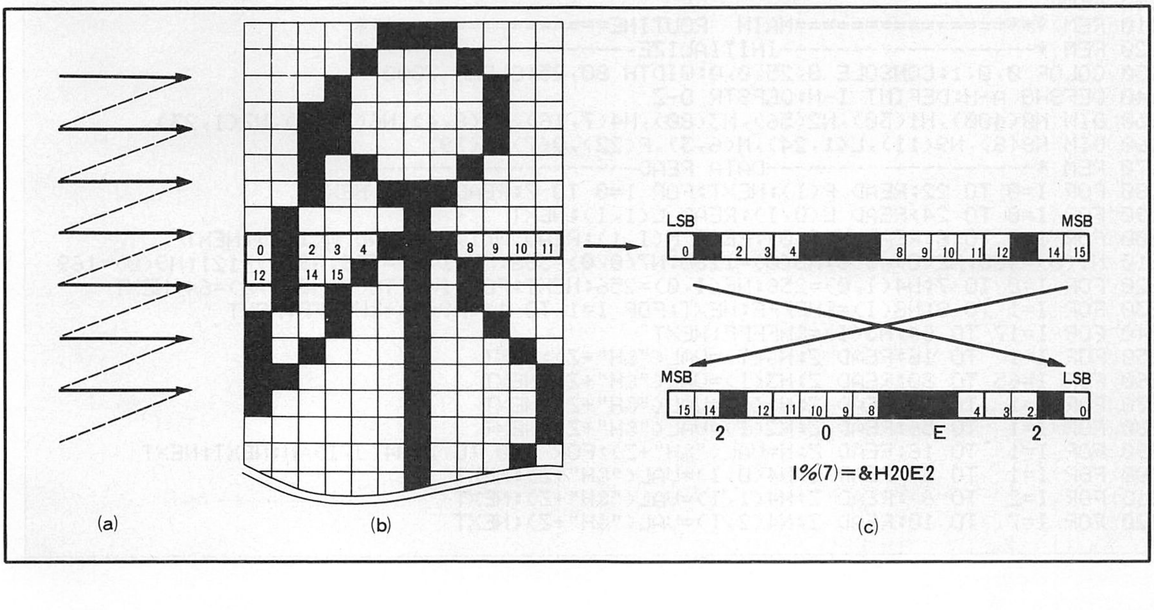
写真19　行番号1330～1500のリスト



③440～480　DEMONSTRATION(PART1)

写真２に示すデモ１です．乱数により切れ目の方向が決定され，それに従って環が表示され，人間の上にカタカナで方向が表示されます．RETURN キーまたは任意の英数キーを押すと⑤へ分岐します．押されない場合には，８回表示を繰り返した後に④のデモ２へ分岐します．

図４　ＰＵＴ＠用グラフィック・データの作成方法



④490〜680　DEMONSTRATION(PART2)

写真3に示すデモ2です．乱数により切れ目の方向が決定され，それに従って環が表示され，対応するキー（図2参照）が反転します．

③と同様に**RETURN**キーまたは任意の英数キーを押すと⑤へ分岐します．押されない場合には，8回表示を繰り返した後に③のデモ1へ分岐します．

⑤690〜980　VISUAL ACUITY CHART DISPLAY

環用のデータを作成し表示します．そして視標数と正答数をカウントし，正答数が規定数に達すれば⑥に分岐し，達しなければ次の段階の環が表示されます．

⑥990〜1080　VISUAL ACUITY DISPLAY

視力を表示させます．ただし，ここでは表示される数字とその位置が設定されるだけで，実際の表示は次の⑦〜⑫で行なわれます．

⑦1090〜1160　"0" DISPLAY

数字『0』の表示です．

⑧1170〜1240　"1" DISPLAY

数字『1』の表示です．

⑨1250〜1320　"2" DISPLAY

数字『2』の表示です．

⑩1330〜1400　"3" DISPLAY

数字『3』の表示です．

⑪1410〜1480　"5" DISPLAY

数字『5』の表示です．

⑫1490〜1500　DECIMAL POINT DISPLAY

小数点の表示です．

4 おわりに

PCのグラフィック・ディスプレイは160ドット×100ドットなので，特に小さな視標ではかなりガタガタな環となりました．やはり，グラフィックは256ドット×256ドットくらいは欲しいですね．

参考文献

1) 大島祐之："視力検査の基準化について"，照明学会雑誌，Vol.49，No.2，1965．
2) 日本電気：PC-8001　USER'S MANUAL, IEM-683A, 1979．
3) 日本電気：PC-8001　N-BASIC REFERENCE MANUAL, IEM-682A, 1979．
4) 飯田・今井・大野・土田：アスキー・システム・バンクPC＃1 N-BASIC入門，アスキー出版，1980
5) 後藤富雄：パーソナル・コンピュータPC-8001のグラフィック機能とその活用法，インターフェース，Vol.5，No.12，1979．

視力検査プログラム・リスト

```
10 REM ***********************************************
20 REM *                                             *
30 REM *          VISUAL ACUITY TEST Ver 1.0         *
40 REM *                                             *
50 REM *         for  NEC PC-8001 BASIC Ver 1.0      *
60 REM *                                             *
70 REM ***********************************************
80 REM *       Copyright (C) 1980  by Ohashi M.      *
90 REM ***********************************************
100 REM
110 REM ***===============MAIN  ROUTINE===============***
120 REM *-------------------INITIALIZE-------------------*
130 COLOR 0,0,1:CONSOLE 0,25,0,0:WIDTH 80,25:CLEAR 7000
140 DEFSNG A-H:DEFINT I-N:DEFSTR O-Z
150 DIM N0(400),N1(30),N2(56),N3(80),N4(7,16),N5(8,4),N6(7,16),N7(1,23)
160 DIM N8(8),N9(11),L(1,24),M(6,3),P(22),Q(7),S(79)
170 REM *-------------------DATA READ-------------------*
180 FOR I=0 TO 22:READ P(I):NEXT:FOR I=0 TO 7:READ Q(I):NEXT
190 FOR I=0 TO 24:READ L(0,I):READ L(1,I):NEXT
200 FOR I=0 TO 6:READ M(I,0):READ M(I,1):READ M(I,2):READ M(I,3):NEXT
210 N1(0)=480:N2(0)=896:N3(0)=1280:N7(0,0)=368:N7(1,0)=368:N8(0)=121:N9(0)=169
220 FOR I=0 TO 7:N4(I,0)=256:N6(I,0)=256:NEXT:FOR I=0 TO 8:N5(I,0)=64:NEXT
230 FOR I=1 TO 8:N8(I)=&HFFFF:NEXT:FOR I=1 TO 11:N9(I)=&HFFFF:NEXT
240 FOR I=17 TO 64:N3(I)=&HFFFF:NEXT
250 FOR I=1  TO 16:READ Z:N3(I)=VAL("&H"+Z):NEXT
260 FOR I=65 TO 80:READ Z:N3(I)=VAL("&H"+Z):NEXT
270 FOR I=1  TO 30:READ Z:N1(I)=VAL("&H"+Z):NEXT
280 FOR I=1  TO 56:READ Z:N2(I)=VAL("&H"+Z):NEXT
290 FOR I=1  TO 16:READ Z:N=VAL("&H"+Z):FOR J=0 TO 7:N4(J,I)=N:NEXT:NEXT
300 FOR I=1  TO 3 :READ Z:N4(0,I)=VAL("&H"+Z):NEXT
310 FOR I=2  TO 6 :READ Z:N4(1,I)=VAL("&H"+Z):NEXT
320 FOR I=7  TO 10:READ Z:N4(2,I)=VAL("&H"+Z):NEXT
```

```
330 FOR I=11 TO 15:READ Z:N4(3,I)=VAL("&H"+Z):NEXT
340 FOR I=14 TO 16:READ Z:N4(4,I)=VAL("&H"+Z):NEXT
350 FOR I=11 TO 15:READ Z:N4(5,I)=VAL("&H"+Z):NEXT
360 FOR I=7  TO 10:READ Z:N4(6,I)=VAL("&H"+Z):NEXT
370 FOR I=2  TO 6 :READ Z:N4(7,I)=VAL("&H"+Z):NEXT
380 FOR I=0 TO 8 :FOR J=1 TO 4:READ Z:N5(I,J)=VAL("&H"+Z):NEXT:NEXT
390 FOR I=1 TO 16:READ Z:N=VAL("&H"+Z):N6(0,I)=N:N6(1,17-I)=N:NEXT
400 FOR I=1 TO 16:READ Z:N=VAL("&H"+Z):N6(2,I)=N:N6(3,17-I)=N:NEXT
410 FOR I=1 TO 16:FOR J=0 TO 3:Y=RIGHT$("000"+HEX$(N6(J,I)),4):Z="":FOR K=1 TO 4
420 Z=LEFT$(P(ASC(MID$(Y,K))-48),1)+Z:NEXT:N6(J+4,I)=VAL("&H"+Z):NEXT:NEXT
430 FOR I=0 TO 1 :FOR J=1 TO 23:READ Z:N7(I,J)=VAL("&H"+Z):NEXT:NEXT
440 REM *-----------DEMONSTRATION (PART 1)-------------*
450 PRINT CHR$(12):PUT@(137,30)-(148,69),N1,PSET:PUT@(16,71)-(143,77),N2,PSET
460 FOR I=1 TO 8:PUT@(0,30)-(31,69),N3,PSET:LOCATE 70,6:PRINT SPC(7):N=RND(1)*8
470 FOR J=0 TO 16:N0(J)=N4(N,J):NEXT:PUT@(8,42)-(23,57),N0,PSET
480 LOCATE 70,6:PRINT Q(N):FOR J=0 TO 100:IF INKEY$<>"" GOTO 690 ELSE NEXT:NEXT
490 REM *-----------DEMONSTRATION (PART 2)-------------*
500 PRINT CHR$(12):FOR I=0 TO 8:L0=L(0,I):L1=L(1,I)
510 LINE(L0,L1)-(L0+10,L1+10),PSET,B:FOR J=0 TO 4:N0(J)=N5(I,J):NEXT
520 PUT@(L0+1,L1+1)-(L0+8,L1+8),N0,PSET:NEXT
530 FOR I=0 TO 7:L0=L(0,I+9):L1=L(1,I+9):FOR J=0 TO 16:N0(J)=N6(I,J):NEXT
540 PUT@(L0,L1)-(L0+15,L1+15),N0,PSET:NEXT
550 FOR I=17 TO 20:L0=L(0,I):L1=L(1,I):PUT@(L0,L1)-(L0+12,L1+12),N9,PSET:NEXT
560 FOR I=0 TO 23:N0(I)=N7(0,I):NEXT
570 FOR I=21 TO 24:L0=L(0,I):L1=L(1,I):PUT@(L0,L1)-(L0+18,L1+18),N0,OR:NEXT
580 FOR I=1 TO 8:N=RND(1)*8:IF (N MOD 2)=1 THEN N=N/2:GOTO 620 ELSE N=N/2
590 FOR J=0 TO 1:L0=L(0,N):L1=L(1,N):PUT@(L0,L1)-(L0+10,L1+10),N8,XOR
600 L0=L(0,N+17):L1=L(1,N+17):PUT@(L0,L1)-(L0+12,L1+12),N9,XOR
610 FOR K=0 TO 500:NEXT:NEXT:GOTO 680
620 FOR J=0 TO 23:N0(J)=N7(1,J):NEXT
630 L0=L(0,N+4):L1=L(1,N+4):PUT@(L0,L1)-(L0+10,L1+10),N8,XOR
640 L0=L(0,N+21):L1=L(1,N+21):PUT@(L0,L1)-(L0+18,L1+18),N0,AND
650 FOR J=0 TO 23:N0(J)=N7(0,J):NEXT
660 L0=L(0,N+4):L1=L(1,N+4):PUT@(L0,L1)-(L0+10,L1+10),N8,XOR
670 L0=L(0,N+21):L1=L(1,N+21):PUT@(L0,L1)-(L0+18,L1+18),N0,OR
680 IF INKEY$<>"" GOTO 690 ELSE NEXT:GOTO 440
690 REM *----------VISUAL ACUITY CHART DISPLAY----------*
700 BEEP:N0(0)=6400:FOR I=0 TO 6:IQ=0:IR=0
710 FOR J=0 TO 79:S(J)=STRING$(80,49):NEXT:N=RND(1)*8+1
720 FOR J=0 TO M(I,1)-M(I,0)-2:K=M(I,0)+1+J:FOR IY=-K TO K:BX=SQR((K-IY)*(K+IY))
730 J0=40-IY:B2=40-BX:B3=39.5+BX:FOR A=0 TO 1 STEP .5
740 MID$(S(J0),B3+A)="0":MID$(S(J0),B2+A)="0"
750 MID$(S(B3+A),J0)="0":MID$(S(B2+A),J0)="0":NEXT:NEXT:NEXT
760 ON N GOTO 840,780,850,800,830,790,820,770
770 J1=-M(I,1):J2=-M(I,0):K1=-M(I,2):K2=+M(I,2):GOTO 810
780 J1=+M(I,0):J2=+M(I,1):K1=-M(I,2):K2=+M(I,2):GOTO 810
790 J1=-M(I,2):J2=+M(I,2):K1=+M(I,0):K2=+M(I,1):GOTO 810
800 J1=-M(I,2):J2=+M(I,2):K1=-M(I,1):K2=-M(I,0)
810 FORJ=40+J1 TO 40+J2:FORK=40+K1 TO 40+K2:MID$(S(J),K)="1":NEXT:NEXT:GOTO 880
820 IX=5:IY=5:IZ=1:GOTO 860
830 IX=40:IY=40:IZ=-1:GOTO 860
840 IX=5:IY=75:IZ=-1:GOTO 860
850 IX=40:IY=40:IZ=1
860 AS=1-SGN(I)*.5:FOR AJ=-M(I,3) TO M(I,3) STEP AS:FOR AK=0 TO 35 STEP AS
870 MID$(S(IY-IZ*AJ+IZ*AK),IX+AJ+AK)="1":NEXT:NEXT
880 N=N-(N=5)*4:PRINT CHR$(12):LINE(35,10)-(125,89),PSET,BF:J1=0
890 FOR J=0 TO 79:W=S(J):FOR K1=1 TO 65 STEP 16:X=MID$(W,K1,16)
900 IF X="1111111111111111" THEN Z="FFFF":GOTO 930 ELSE Z=""
910 FOR K2=1 TO 13 STEP 4:K3=VAL(MID$(X,14-K2,4))
920 Z=Z+HEX$(-(K3\1000)*19-(K3\100)*38-(K3\10)*76+K3*8):NEXT
930 J1=J1+1:N0(J1)=VAL("&H"+Z):NEXT:NEXT
940 PUT@(40,10)-(119,89),N0,PSET:BEEP
950 Z=INKEY$:IF Z="" GOTO 950 ELSE BEEP:IF Z="5" GOTO 980 ELSE IQ=IQ+1
960 IF " "+Z=STR$(N) THEN IR=IR+1
970 IF IQ<4 GOTO 710 ELSE IF IR<3 GOTO 980 ELSE GOTO 990
980 NEXT:GOTO 400
990 REM *------------VISUAL ACUITY DISPLAY------------*
1000 WIDTH 40,25:ON I+1 GOSUB 1020,1030,1040,1050,1060,1070,1080:GOSUB 1490
1010 FOR I=0 TO 5000:NEXT:WIDTH 80,25:GOTO 440
```

●視力検査プログラム・リスト

```
1020 I=7:GOSUB 1170:I=23:GOSUB 1090:I=19:RETURN
1030 I=7:GOSUB 1090:I=23:GOSUB 1410:I=19:RETURN
1040 I=7:GOSUB 1090:I=23:GOSUB 1330:I=19:RETURN
1050 I=1:GOSUB 1090:I=17:GOSUB 1250:I=29:GOSUB 1410:I=13:RETURN
1060 I=7:GOSUB 1090:I=23:GOSUB 1250:I=19:RETURN
1070 I=1:GOSUB 1090:I=17:GOSUB 1170:I=29:GOSUB 1410:I=13:RETURN
1080 I=7:GOSUB 1090:I=23:GOSUB 1170:I=19:RETURN
1090 REM ***==========="0" DISPLAY ROUTINE===========***
1100 LOCATE I,6 :PRINT "           ":LOCATE I,7 :PRINT "           "
1110 LOCATE I,8 :PRINT "           ":LOCATE I,9 :PRINT "           "
1120 LOCATE I,10:PRINT "           ":LOCATE I,11:PRINT "           "
1130 LOCATE I,12:PRINT "           ":LOCATE I,13:PRINT "           "
1140 LOCATE I,14:PRINT "           ":LOCATE I,15:PRINT "           "
1150 LOCATE I,16:PRINT "           ":LOCATE I,17:PRINT "           "
1160 LOCATE I,18:PRINT "           ":LOCATE I,19:PRINT "           ":RETURN
1170 REM ***==========="1" DISPLAY ROUTINE===========***
1180 LOCATE I,6 :PRINT "           ":LOCATE I,7 :PRINT "           "
1190 LOCATE I,8 :PRINT "           ":LOCATE I,9 :PRINT "           "
1200 LOCATE I,10:PRINT "           ":LOCATE I,11:PRINT "           "
1210 LOCATE I,12:PRINT "           ":LOCATE I,13:PRINT "           "
1220 LOCATE I,14:PRINT "           ":LOCATE I,15:PRINT "           "
1230 LOCATE I,16:PRINT "           ":LOCATE I,17:PRINT "           "
1240 LOCATE I,18:PRINT "           ":LOCATE I,19:PRINT "           ":RETURN
1250 REM ***==========="2" DISPLAY ROUTINE===========***
1260 LOCATE I,6 :PRINT "           ":LOCATE I,7 :PRINT "           "
1270 LOCATE I,8 :PRINT "           ":LOCATE I,9 :PRINT "           "
1280 LOCATE I,10:PRINT "           ":LOCATE I,11:PRINT "           "
1290 LOCATE I,12:PRINT "           ":LOCATE I,13:PRINT "           "
1300 LOCATE I,14:PRINT "           ":LOCATE I,15:PRINT "           "
1310 LOCATE I,16:PRINT "           ":LOCATE I,17:PRINT "           "
1320 LOCATE I,18:PRINT "           ":LOCATE I,19:PRINT "           ":RETURN
1330 REM ***==========="3" DISPLAY ROUTINE===========***
1340 LOCATE I,6 :PRINT "           ":LOCATE I,7 :PRINT "           "
1350 LOCATE I,8 :PRINT "           ":LOCATE I,9 :PRINT "           "
1360 LOCATE I,10:PRINT "           ":LOCATE I,11:PRINT "           "
1370 LOCATE I,12:PRINT "           ":LOCATE I,13:PRINT "           "
1380 LOCATE I,14:PRINT "           ":LOCATE I,15:PRINT "           "
1390 LOCATE I,16:PRINT "           ":LOCATE I,17:PRINT "           "
1400 LOCATE I,18:PRINT "           ":LOCATE I,19:PRINT "           ":RETURN
1410 REM ***==========="5" DISPLAY ROUTINE===========***
1420 LOCATE I,6 :PRINT "           ":LOCATE I,7 :PRINT "           "
1430 LOCATE I,8 :PRINT "           ":LOCATE I,9 :PRINT "           "
1440 LOCATE I,10:PRINT "           ":LOCATE I,11:PRINT "           "
1450 LOCATE I,12:PRINT "           ":LOCATE I,13:PRINT "           "
1460 LOCATE I,14:PRINT "           ":LOCATE I,15:PRINT "           "
1470 LOCATE I,16:PRINT "           ":LOCATE I,17:PRINT "           "
1480 LOCATE I,18:PRINT "           ":LOCATE I,19:PRINT "           ":RETURN
1490 REM ***======DECIMAL POINT DISPLAY ROUTINE=======***
1500 LOCATE I,18:PRINT "  ":LOCATE I,19:PRINT "  ":RETURN
1510 END
1520 REM ***=============DATA DEFINITION=============***
1530 REM *---------------CHARACTER DATA----------------*
1540 DATA 0,8,4,C,2,A,6,E,1,9,0,0,0,0,0,0,5,D,3,B,7,F     :'P
1550 DATA ｳｴｰ,ﾐｷﾞｳｴｰ,ﾐｷﾞｰ,ﾐｷﾞｼﾀｰ,ｼﾀｰ,ﾋﾀﾞﾘｼﾀｰ,ﾋﾀﾞﾘｰ,ﾋﾀﾞﾘｳｴｰ :'Q
1560 REM *---------------POSITION DATA-----------------*
1570 DATA 74,31,87,44,74,57,61,44,87,31,87,57,61,57,61,31,74,44        :'L
1580 DATA 57,10,57,73,40,27,40,56,86,10,86,73,103,27,103,56
1590 DATA 73,10,106,43,73,76,40,43,93,17,93,63,47,63,47,17
1600 DATA 1,3,0,0,2,4,1,1,5,9,1,1,5,10,1,2,7,13,2,2,9,17,3,3,14,25,4,4 :'M
1610 REM *---------------GRAPHIC DATA------------------*
1620 DATA 0007,0000,03FF,0000,FFFF,0000,FFFF,003F       :'N3(1-16)
1630 DATA FFFF,0FFF,FFFF,FFFF,FFFF,FFFF,FFFF,FFFF
1640 DATA FFFF,FFFF,FFFF,FFFF,FFFF,FFFF,FFFF,0FFF       :'N3(65-80)
1650 DATA FFFF,003F,FFFF,0000,03FF,0000,0007,0000
1660 DATA 00E0,0811,20C2,420C,E411,0420                 :'N1
1670 DATA 20E2,4A15,2491,3445,4184,8408
1680 DATA 0440,4044,2402,0140,6014,0600
1690 DATA 8050,4804,0440,1042,4104,0410
1700 DATA 4042,4404,0480,0050,4004,0400
```

```
1710 DATA 0008,0000,0000,7C00,0000,0000,0000,1000        :'N2
1720 DATA 0004,0000,0000,0400,0000,0000,0000,2000
1730 DATA 0002,0000,0000,3C00,0016,0000,0000,4000
1740 DATA FFFF,FFFF,FFFF,40FF,FF2A,FFFF,FFFF,FFFF
1750 DATA 0002,0000,0000,4000,002A,0000,0000,4000
1760 DATA 0004,0000,0000,4400,002A,0000,0000,2000
1770 DATA 0008,0000,0000,3800,002A,0000,0000,1000
1780 DATA F81F,E007,C003,83C1,87E1,0FF0,1FF8,1FF8        :'N4(0-7)
1790 DATA 1FF8,1FF8,0FF0,87E1,83C1,C003,E007,F81F
1800 DATA FBDF,E3C7,C3C3,F007,F803,FFC1,BFE1,1FF0        :'N4(0,1)
1810 DATA FFF8,FFF8,FFF8,FFF8,1FF0,BFE1,FFC1,F803,F007   :'N4(2,3)
1820 DATA C3C3,E3C7,FBDF,0FF8,87FD,83FF,C01F,E00F        :'N4(4,5)
1830 DATA 1FFF,1FFF,1FFF,1FFF,E00F,C01F,83FF,87FD,0FF8   :'N4(6,7)
1840 DATA 3800,4444,4438,3844,3000,0408,443C,3844,3800,4044,1020,7C08:'N5(8,6,2)
1850 DATA 2000,2830,7C24,2020,3800,4444,4078,1820,7C00,1020,4020,3844:'N5(4,9,3)
1860 DATA 1000,1018,1010,3810,7C00,2040,0810,0808,7C00,3C04,4040,3844:'N5(1,7,5)
1870 DATA 0000,F000,FE00,FFC0,FFF0,FFFC,FFFF,FFFF        :'N6(0-3)
1880 DATA FFFE,FFFC,FFF8,FFF0,FFE0,FFC0,1F80,0700
1890 DATA 00C0,01C0,03E0,07E0,0FF0,1FF0,3FF8,7FF8        :'N6(4-7)
1900 DATA FFF8,FFFC,FFFC,7FFC,7FFE,3FFE,3FFE,3FFE
1910 DATA 0200,3800,E000,8003,003F,03FE,3FF8,FFE0,FF83,FE3F,FBFF     :'N7(0)
1920 DATA BFFF,FFFF,FFF8,FF83,F80F,803F,00FF,03F8,0F80,3800,8000,0000
1930 DATA FDFF,C7FF,1FFF,7FFC,FFC0,FC01,C007,001F,007C,01C0,0400     :'N7(1)
1940 DATA 4000,0000,0007,007C,07F0,7FC0,FF00,FC07,F07F,C7FF,7FFF,FFFF
```

医用エレクトロ絵本

テレビ・アニメ

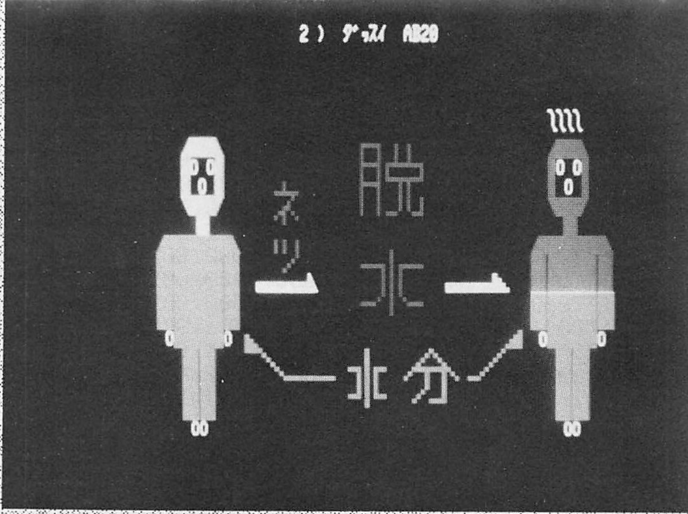


写真1　脱水症状の説明

中部マイコンクラブ医用マイコン研究会　近藤　亨

患者さんの診察で『病気の説明』は大切なことですが，時間がかかって難儀な場合がよくあります．こんなときに話の他に絵や表を見せることができれば，理解が早まり診察のスピードアップに役立つし，誤解防止の有力な方法になります．

そこで試みたのがPC-8001を使ったテレビ・マンガで，考え方はテレビの画面に絵や表を書き，マシン語を使って再生するという簡単な方法です．しかし，効果はバツグンなので発表します．

写真2　心蔵の環状動脈の説明

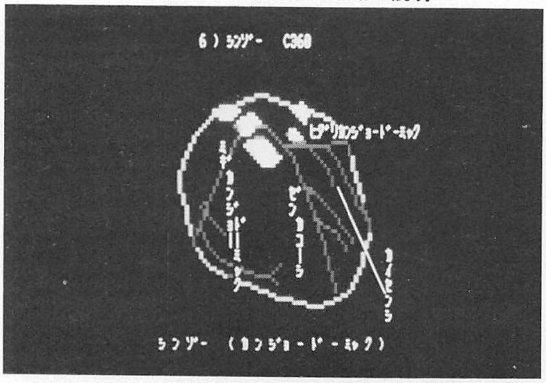
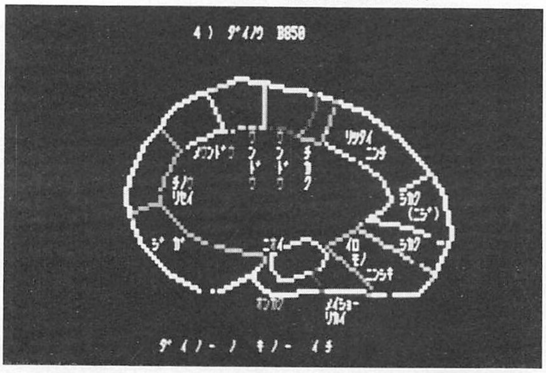


写真3　大脳の機能位置の説明



1.システム

システムPC-8001（RAMフル実装）とカラーモニタPC-8043で構成しています．データはカセットに入れて使っていますが，将来はミニフロッピーを予定しています．

2.プログラムの要点

前述のようにBASICプログラムで描いてマシン語で再生しています．描く場合はどうせ考えながらですから，ゆっくり描けば良いでしょう．したがって，BASICプログラムですから，メモリの余裕がある限り色々と便利な手順を付け加えていくことができます．再生時のマシン語プログラムは画面再現の時間が早く，パッパッと思い通り絵が出てきます．

こんな観点から，ここではとりあえず次のような仕様を考えてみました．

❶絵はドット・グラフィックとキャラクタ・コード表にある各シンボルが併用できること．
❷白黒，カラーが任意であること．
❸アニメ作製のため，現われている画像の上に，さらに追加，書き加え，変更ができること．
❹画面作製時にタイトルを書くことで，再生時に内容のリストが一覧表となって表示できること．
❺再生には必要な画面と再現枚数が指定できること．
❻再生時に気に入った画面を選んで，それ以後その画面をもとに追加，変更などの操作ができること．
❼画像再生は瞬時であること．
❽できるだけ多くの画面を入れるため，メモリの節約を図ること．

3.プログラムの概要

BASICプログラムのフローチャートを図1に示します．



図1(a) BASICのフローチャート

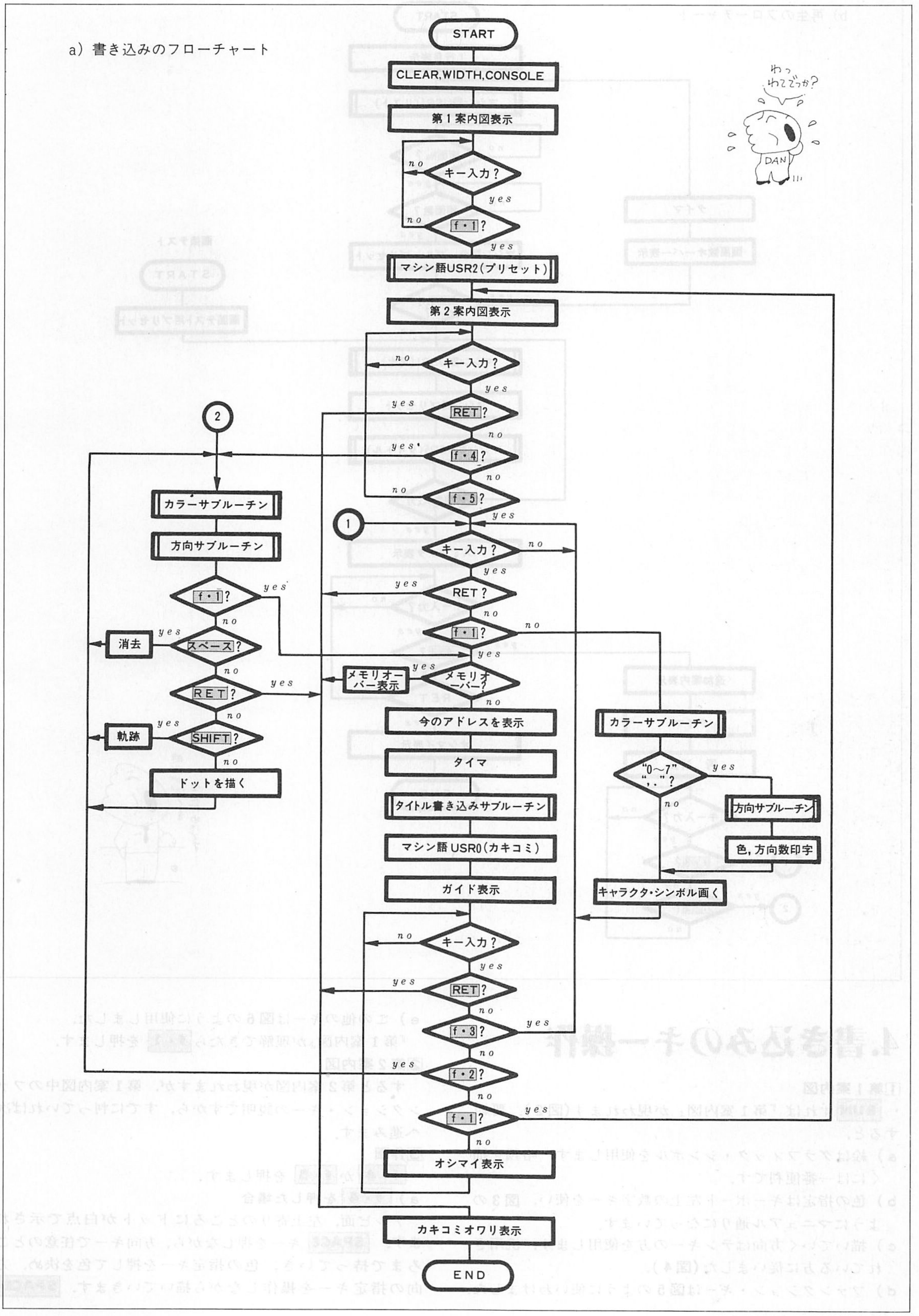
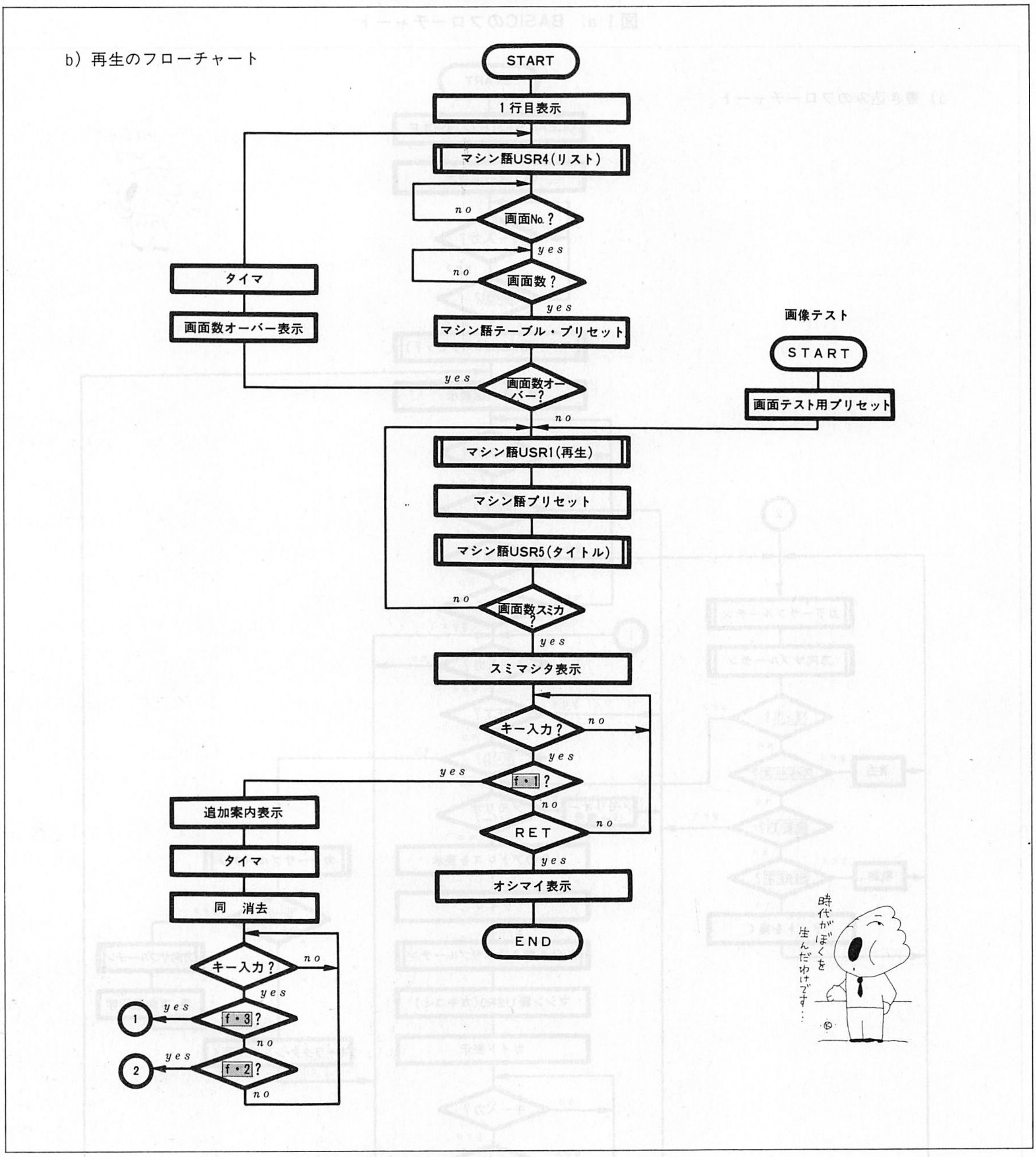


図1(b)　BASICのフローチャート



4.書き込みのキー操作

1第1案内図

RUN すれば『第1案内図』が現われます(図2)．要約すると，

a）絵はグラフィック・シンボルを使用します．略画を描くには一番便利です．

b）色の指定はキーボード左上の数字キーを使い，図3のようにマニュアル通りになっています．

c）描いていく方向はテンキーの方を使用します．汎用されている方に従いました(図4)．

d）ファンクション・キーは図5のように使いわけました．

e）この他のキーは図6のように使用しました．

『第1案内図』が理解できたら **f・1** を押します．

2第2案内図

すると第2案内図が現われますが，第1案内図中のファンクション・キーの説明ですから，すでに判っていれば次へ進みます．

3作画

f・4 か **f・5** を押します．

a）**f・4** **を押した場合**

テレビ画，左上寄りのところにドットが白点で示されます． **SPACE** キーを押しながら，方向キーで任意のところまで持っていき，色の指定キーを押して色を決め，方向の指定キーを操作しながら描いていきます． **SPACE**

図2　書き込みの手順

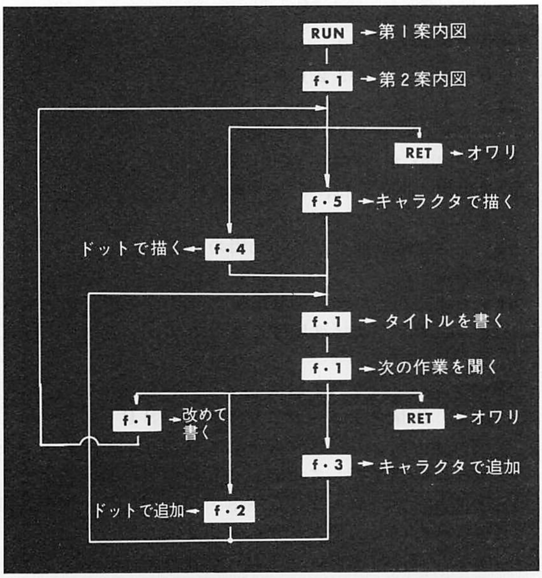


図3　色の指定

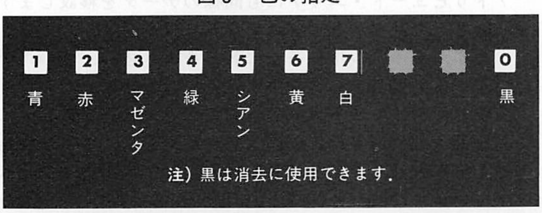
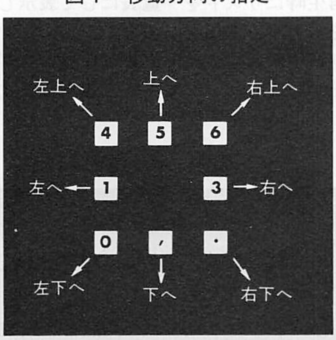


図4　移動方向の指定





キーと SHIFT キーを使い分けて，描いていけば任意の図柄が描けます．

SHIFT キーは消去しないで軌跡を描いていくので，ドットの追跡に便利です．

B）f・5 を押した場合

まず，画面中央に白でnと出てきます(nはrun の最後の文字です)．グラフィック・スイッチを押しながら任意のキャラクタ・シンボルを押すと，同じく白でnがシンボルの図柄に変わります．次に色の指定キーを押すと色が指定の色に変わります．

そこで，方向の指定キーを使って描くのですが，グラフィックの場合と異なり方向キーを1回押すと，1つだけ同一のキャラクタ・シンボルが描かれます．

このようにしたのは使い勝手が良かったからで，変更も自由にできます．なお，色，方向ともに押したキーの英数文字が一瞬現われます．特に消去時（スペース・キー使用時のこと）の位置確認に便利です．

④タイトル付け

描き終ったら f・1 を押します．すると，画面上部に画面番号と今描き終った絵を，書き込むべきメモリのスタート番地が現われ，数秒後に中央の一部が消えます．そこへタイトル名を書き込み，次に f・1 を押します．

⑤次の作業は…

すると，次の作業を聞いてきます．それに従って f・1，f・2，f・3，あるいは RET キーを押してください．

ただし，押し違えると STOP キー以外は押し違いを表示するため，正しく押し直してください．STOP キーを押せばBASICプログラムは中断します．

⑥新画面にするには

f・1 を押せば『第2案内図』を表示します．新しく別の絵を書く場合は，続いて f・4 あるいは f・5 を押します．

⑦旧画面にドットで追加するとき

f・2 を押すと，今の画像がそのまま残り画面最上部に案内と共に左上方部にドットが白くポツンと出ます．したがって，このドットを使って画面に追加，あるいは変更を加えていくことができます．

⑧旧画面にキャラクタで追加するとき

f・3 を押した場合は，画面中央部にｇｏｔｏの文字が同一場所に次々現われては消えます．消えたらキャラクタ・シンボルを使って画面に追加，変更を加えるわけです．

⑨終了したいとき

RET を押すと今の画面をメモリに書き込んで『書き込み終り』を表示し，プログラムを終わります．

⑩追加が終って

⑦，⑧でそれぞれ絵の追加，変更を終えたら f・1 を押します．

⑪追加のタイトル

図5　ファンクション・キーの指定

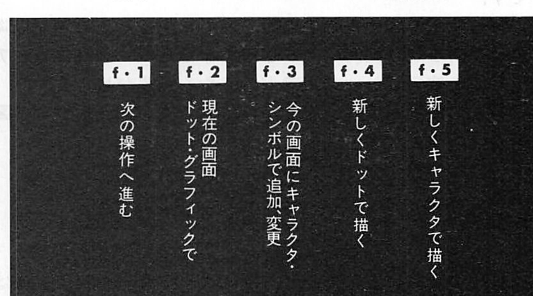
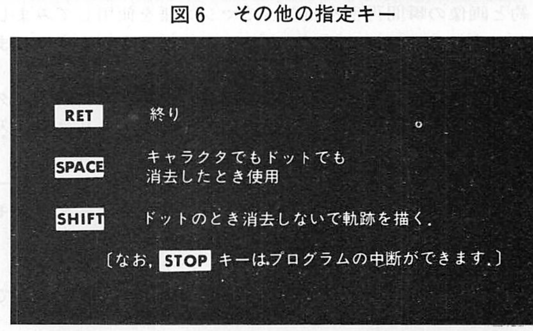


図6　その他の指定キー



画面上部に4と同様にタイトルを書き，f・1を押します．

12継続

以下同様に，新しい画面に移るのなら再びf・1を，追加するのならf・2　f・3を，中止および終了するならRETを押します．こうして次々と書き続けていくわけです．

13メモリ・オーバー

メモリの余裕がなくなると『メモリ・オーバー』を告げ，その画面をメモリへ書き込んだうえ，終了します．

5.再生のキー操作

1画面テスト

メモリに書き入れた全部の画面をテストしてみる場合で，f・9（SHIFTキーとf・4を同時に押す）を押すと，入れた画像を最初から最後までゆっくり順番に再生できます．

2画面再生

f・8（SHIFTキーとf・3を同時に押す）を押すと入っている画面のリスト一覧表が現われます．そこで希望の画面ナンバーを入力し，次にその画面から何場面出すかを入力します．

すると，まず希望の画面が現われ，続いて入力した画面数が終るまで順次画面を表示します．ない場面まで求められると『オーバー』を現わして，再びリストを出します．したがって，押し直してください．

3終了

1，2とも終ればRETを押して終了します．

4追加変更

再生時にも画面の追加変更が可能です．これを行なうには，その画面を呼び出して（画面数は1を入力する）RETの代りにf・1を押します．

すると，f・2かf・3かを尋ねてきますから〔書き込み〕時と同様，それぞれf・2，f・3を使いわけて押し，ドットまたはキャラクタ・シンボルで書き加えていくわけです．

特に注意を要するのは追加，変更をするとその画面以降の呼び出しが自動的にはできなくなるため，画面番号とアドレスを控えておいて手で入れてやる必要が起きます．

再生時の手順は以上のように簡単ですが，表示すると図7のようになります．

6.マシン語プログラム

GET@，PUT@などのN-BASIC言語を利用すれば画像の扱いは簡単ですが，最初に述べたようにメモリの節約と画像の瞬間再現をねらってマシン語を使用してみました．そして，マシン語プログラムの仕様は次のようにしました．

❶ビデオRAMエリア上の絵の位置と大きさを判断し，タテヨコ必要最少限の大きさを決めて画素をメモリに取り入れます．

❷アトリビュート・エリア上の，変化を起こしているデータの範囲[注)]を探して幅を決め，この範囲内のデータをメモリに取り入れます．

❸メモリ上（A000～A00F）にテーブルを設定し，プログラムの作製を簡単にし，通常の使用法（書き込んでから引き続き再生する場合）では，BASICプログラムに続いてマシン語プログラム分だけ（データ分の領域は不要）をカセット・テープへ移せば良いようにしました（A000番地からA32F番地まで）．

❹タイトル並みにタイトル一覧リスト用にも同様テーブル（9FF0～9FFF）を使いました(図8)．

❺『メモリ・オーバー』，『画面数オーバー』などのメッセージ表示用．あるいは，テーブルのプリセット用などにもそれぞれマシン語プログラムを利用しました．

❻マシン語は原則として8080用のを用い，一部Z80用のを追加しました．

❼メモリ節約とアニメ効果増大にはさらに良い方法もあると思いますが今回は汎用性を考えてプログラムを決定しました．

❽マシン語プログラムの構成（リスト参照）

No.1　画像ストア・プログラム（USR0）

前に述べたようにBASICプログラムで画いた画像のタテヨコの幅を探して決め，必要最少限の領域をメモリへ書き込みます．アトリビュート・エリアについても同様に必要な幅だけの分をメモリへ蓄えます．

No.2　画像再生プログラム（USR1）

蓄えたメモリの内容をビデオRAMエリア上に再現し，アトリビュート・エリアにも同様のデータを移設します．

No.3　プリセット・プログラム（USR2）

まず，タイトル用メモリにFFを書き込み，2つのテーブルにも必要なデータを書き込んでおきます．

No.4　タイトル・ストア・プログラム（USR3）

画面上端の第1行目に表示された画面No.，メモリ開始番地と書き入れたタイトル名の16文字分を順次にメモリへ蓄えていきます．

No.5　タイトル・リスト表示プログラム（USR4）

画像再生時にリストを一覧表にして表示し，画面No.や再生画数を入力する準備をします．

No.6　再生画面にタイトルを付けるプログラム(USR5)

画面上端の1行目にタイトルを再現します．

図7　再生の手順

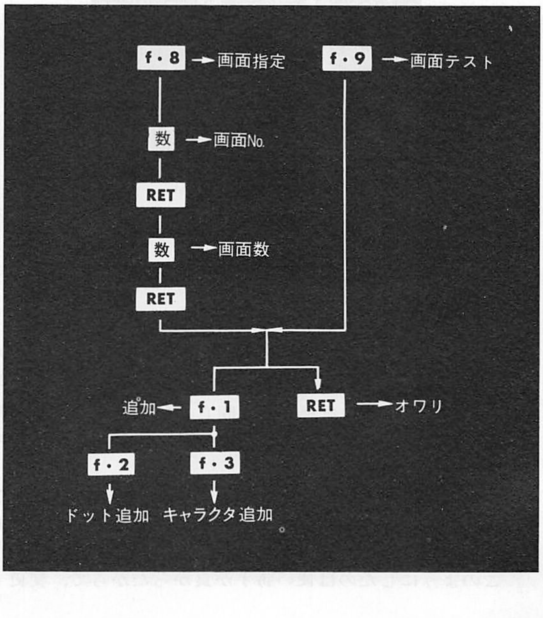


図8　画像的およびタイトル・リスト用テーブル

画像用テーブル

A000	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
画像ストア用メモリスタート・アドレス		アトリビュートスタート・アドレス		ビデオRAM上画スタート・アドレス		画面タテ行数	画面ヨコ幅	アトリ幅					今書き込んだメモリ最終アドレス		FF
下位	上位	下位	上位	下位	上位								下位	上位	

タイトル・リスト用テーブル

9FF0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
タイトル用メモリスタート・アドレス		タイトルNo.	リスト・メモリアドレス用		タイトル・ストアメモリ・アドレス		画面数オーバーチェック		予備						

7. その他

画像再生時のＢＡＳＩＣプログラムに少し手を加えると画の繰り返し再生ができ，アニメ効果が向上します．その一例をプログラム・リストで示します(**リスト４**)．

これは**RUN**すると，画面内容のリストを表示します．そこで，まず希望のスタート画面No.を入力し，続いて反復する画面数を入力，最後に繰り返し回数を入力します．すると，アニメ式に希望の画面が繰り返し再現できます．画面内容の全部は**f・9**を押せばゆっくり出てくるので，必要画面をあらかじめ見ることができます．

プログラムの作制には『書き込みプログラム』で作画を終ったら，そこへテープからこの『編集プログラム』をロードするだけでできます．ロードすると『書き込みプログラム』が『編集プログラム』に入れ替り，マシン語プログラムおよびデータがそのまま残っているので，すぐRUNできます．

以上で説明を終わります．ＰＣ-8001もＴＫ-80ＢＳのようにハード的なことまでもう少しわかると良いのですが，乏しい資料と手探り式の勉強で一応こんなことをやってみました．実際に使ってみて患者さんに好奇の目を見張らせながら，病気を理解して貰うことができ，趣味と実益の両得をしたような良い気分で診察がはかどる昨今です．

◘参考文献

1）NEC：PC-8001 USER'S MANUAL

2）NEC：PC-8001 N-BASIC REFERENCE MANUAL

3）N-BASIC入門，アスキー出版

写真5　ドットで描いたときの画面

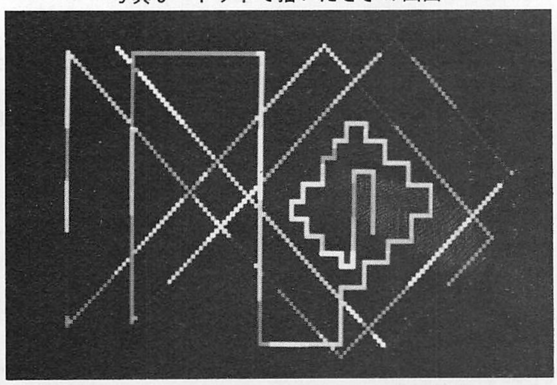


写真6　キャラクタで描いたときの画面

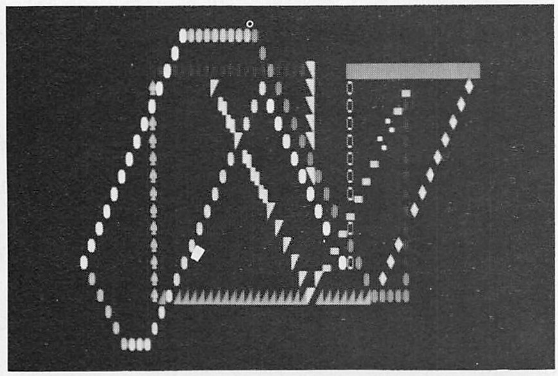


写真4　RUN直後に表示される第１案内図

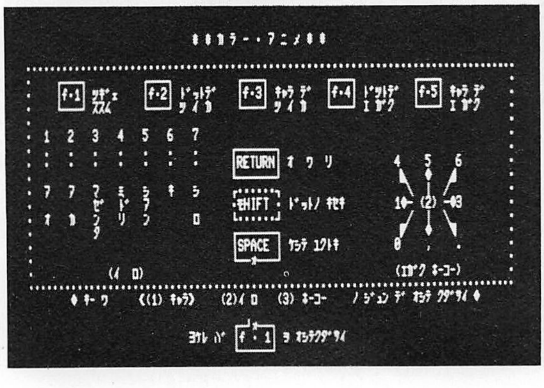


写真7　アニメーション例　体操その１

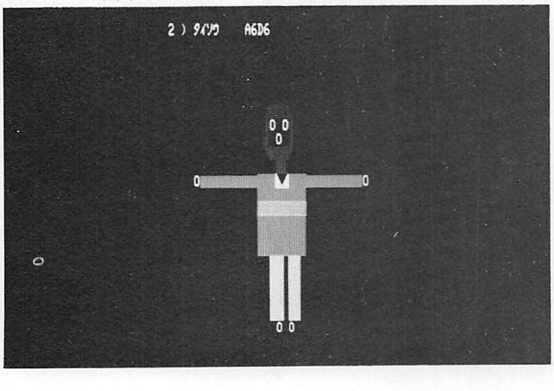


写真８　アニメーション例　体操その２

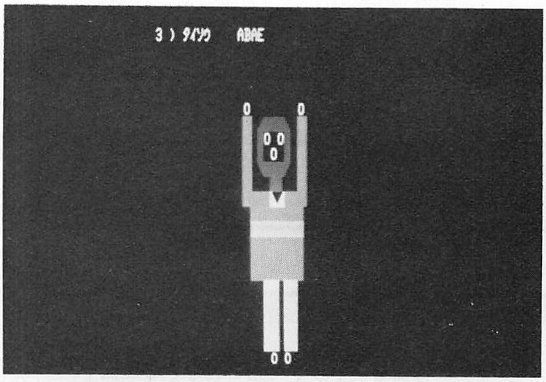
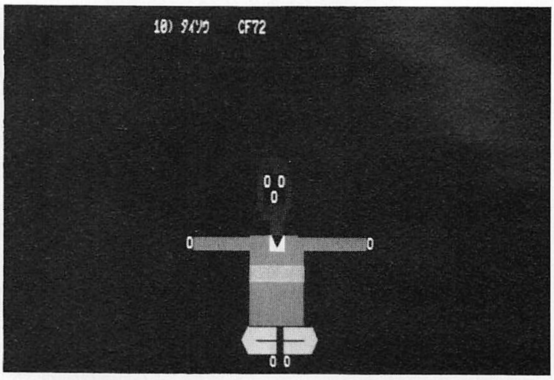


写真９　アニメーション例　体操その３



書き込み用プログラム・リスト

```
10 CLEAR 500,&H9D9F:WIDTH80,25:CONSOLE,,0,1:KEY8,"goto1110"+CHR$(13):KEY9,"goto1
200"+CHR$(13)

20 PRINTCHR$(12):LOCATE27,0:PRINT"* * ｶ ﾗ ｰ ･ ｱ ﾆ ﾒ * *"

30 LOCATE5,3:PRINT"┌────┐          ┌────┐          ┌────┐          ┌────┐          ┌────
┐"

40 LOCATE5,4:PRINT"|f･1| ﾂｷﾞｪ     |f･2| ﾄﾞｯﾄﾃﾞ    |f･3| ｷｬﾗ ﾃﾞ   |f･4| ﾄﾞｯﾄﾃﾞ   |f･5
| ｷｬﾗ ﾃﾞ"

50 LOCATE5,5:PRINT"└────┘ ｽｽﾑ     └────┘ ﾂ ｲ ｶ    └────┘ ﾂ ｲ ｶ    └────┘ ｴ ｶﾞｸ   └────
┘ ｴ ｶﾞｸ"

60 LOCATE3,7:PRINT"1   2   3   4   5   6   7"

70 LOCATE3,8:PRINT".   .   .   .   .   .   .":LOCATE33,8:PRINT"┌────────┐"

80 LOCATE3,9:PRINT"･   .   .   .   .   .   ･":LOCATE33,9:PRINT"|RETURN| ｵ  ﾜ  ﾘ"
:LOCATE59,9:PRINT"4    5    6"

90 LOCATE33,10:PRINT"└────────┘":LOCATE60,10:PRINT"◣   ◆   ◢"

100 LOCATE3,11:PRINT"ｱ   ｱ   ﾏ   ﾐ   ｼ   ｷ   ｼ":LOCATE33,11:PRINT"┌･･･････┐":LOCA
TE61,11:PRINT"＼  |  ／"

110 LOCATE11,12:PRINT"ｾﾞ  ﾄﾞ  ｱ":LOCATE33,12:PRINT"･SHIFT ･ ﾄﾞｯﾄﾉ ｷｾｷ":LOCATE59,
12:PRINT"1◆- (2) -◆3"

120 LOCATE3,13:PRINT"ｵ   ｶ   ﾝ   ﾘ   ﾝ       ﾛ":LOCATE33,13:PRINT"└･･････┘":LOCA
TE61,13:PRINT"／  |  ＼"

130 LOCATE11,14:PRINT"ﾀ":LOCATE33,14:PRINT"┌────────┐":LOCATE60,14:PRINT"◤   ◆   ◥
"

140 LOCATE33,15:PRINT"|SPACE | ｹｼﾃ ﾕｸﾄｷ":LOCATE59,15:PRINT"0     ,     ."

150 LOCATE33,16:PRINT"└────────┘"

160 LOCATE13,17:PRINT"(ｲ  ﾛ)":LOCATE59,17:PRINT"(ｴｶﾞｸ ﾎｰｺｰ)"

170 LOCATE7,19:PRINT"◆ ｷｰ ﾜ     <(1) ｷｬﾗ>    (2)ｲ ﾛ   (3) ﾎｰｺｰ    ﾉ ｼﾞｭﾝ ﾃﾞ ｵｼﾃ
ｸﾀﾞｻｲ ◆"

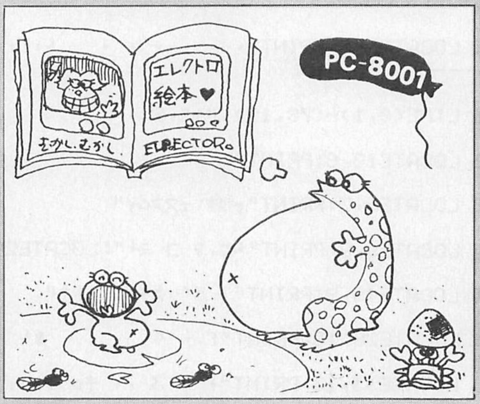
180 LOCATE33,21:PRINT"┌────────┐"

190 LOCATE26,22:PRINT"ﾖｹﾚ ﾊﾞ |f ･ 1| ｦ ｵｼﾃｸﾀﾞｻｲ"

200.LOCATE33,23:PRINT"└────────┘"

210 LINE(0,2)-(78,18),"･",B
```

```
220 Z$=INKEY$:IF Z$=""THEN 220
230 E=INP(9):IF E<>253 THEN 220
240 K=0:Q=0:DEFUSR2=&HA240:X=USR2(Y)
250 PRINTCHR$(12):Q=1
260 LOCATE5,3:PRINT"・ ・ ・ ・ ・ ・   ┌・・・・・・・・・┐   ┌・・・・・・・・・┐   ┌──────────┐   ┌──
────┐."
270 LOCATE5,4:PRINT"・   f・1   ・   ・   f・2   ・   ・   f・3   ・   │   f・4   │   │
f・5   │"
280 LOCATE5,5:PRINT"・ ・ ・ ・ ・ ・   └・・・・・・・・・┘   └・・・・・・・・・┘   └──────────┘   └──
────┘"
290 LINE(0,1)-(78,19),"+",B
300 LOCATE10,6:PRINT"◆"
310 LOCATE6,7:PRINT"┌ﾂｷﾞｪｽｽﾑ┐"
320 LOCATE6,8:PRINT"└ｶ ｷ ｺ ﾐ┘":LOCATE20,8:PRINT"f ・ 5         ●ｷｬﾗｸﾀ ｸﾞﾗﾌｨｯｸ ┐"
330 LOCATE48,9:PRINT"ﾄ ﾃﾞ ｶｷｺﾑ  ﾄｷ"
340 LOCATE20,10:PRINT"f ・ 4         ●ﾄﾞｯﾄ  ｸﾞﾗﾌｨｯｸ ┘"
350 LOCATE33,13:PRINT"f ・ 3 ・・ ｷｬﾗｸﾀ ┐"
360 LOCATE14,14:PRINT"ﾓ ｼ   2 ｶｲ ﾒ ｶﾗ                    ├ ﾃﾞ ﾂｲｶ ｽﾙﾄｷ"
370 LOCATE33,15:PRINT"f ・ 2 ・・ ﾄﾞｯﾄ ┘"
380 LOCATE14,17:PRINT"ｵ ﾜ ﾘ  ﾀｲ ﾄｷ      RETURN    ｦ   ｵｼﾃ   ｸﾀﾞｻｲ"
390 LOCATE6,21:PRINT"♠ ♥ ♣ ｷｬﾗ (1) ｷｬﾗ  (2) ｲ ﾛ  (3) ﾎｰｺｰ  ﾉ ｼﾞｭﾝ ﾃﾞ  ｵｼﾃ ｶｲ
ﾃ ｸﾀﾞｻｲ"
400 LOCATE6,23:PRINT"-・──ﾄﾞｯﾄ (1) ｲﾛ  (2) ﾎｰｺｰ  ﾄ. (3) ｽﾍﾟｰｽ(ｹｽ)  (4) ｼﾌﾄ(ｷｾｷ)
ﾓ ﾂｶｳ"
410 Z$=INKEY$:IF Z$=""THEN 410
420 R=INP(1):IF R=127 THEN 870
430 E=INP(9):IF E<>239 THEN 440 ELSE GOTO 900
440 IF E<>223 THEN 410
450 PRINTCHR$(12):GOTO 460
460 D=7:X=40:Y=13
470 A$=INKEY$:IF A$="" THEN 470
480 R=INP(1):IF R=127 THEN 870
490 S=INP(0):E=INP(9):IF E=253 THEN 770
500 C=INP(6):GOSUB 670
510 IF A$="0" THEN 550 ELSE IF A$="1" THEN 550 ELSE IF A$="2" THEN 550 ELSE IF A
$="3" THEN 550 ELSE IF A$="4" THEN 550
520 IF A$="5" THEN 550 ELSE IF A$="6" THEN 550 ELSE IF A$="7" THEN 550 ELSE IF A
$="," THEN 550 ELSE IF A$="." THEN 550
530 B$=A$
540 LOCATE X,Y:PRINTB$:GOTO 470
```




```
550 GOSUB 570
560 LOCATE X,Y:PRINT A$:FOR I=1 TO 50:NEXT I:GOTO 540
570 IF S=255 THEN X=X:Y=Y
580 IF S=254 THEN X=X-1:Y=Y+1
590 IF S=253 THEN X=X-1
600 IF S=239 THEN X=X-1:Y=Y-1
610 IF S=223 THEN Y=Y-1
620 IF S=191 THEN X=X+1:Y=Y-1
630 IF S=247 THEN X=X+1
640 IF R=223 THEN Y=Y+1
650 IF R=191 THEN X=X+1:Y=Y+1
660 RETURN
670 IF C=255 THEN D=D
680 IF C=254 THEN D=0
690 IF C=253 THEN D=1
700 IF C=251 THEN D=2
710 IF C=247 THEN D=3
720 IF C=239 THEN D=4
730 IF C=223 THEN D=5
740 IF C=191 THEN D=6
750 IF C=127 THEN D=7
760 COLOR D,0,1:RETURN
770 L=PEEK(&HA00E):M=PEEK(&HA00D):IF L>=226 THEN 890
780 K=K+1:COLOR 7:LOCATE28,0:PRINTK:LOCATE31,0:PRINT")ｲﾏﾉ ｱﾄﾞﾚｽ"HEX$(L)HEX$(M):F
OR M=1 TO 800:NEXT M
790 LOCATE32,0:PRINT"             ":LOCATE46,0:PRINT"  ｷｴﾀ ﾄｺﾛ ｪ ﾀｲﾄﾙ ｦ ｶｸｺﾄ       ":
GOSUB 1020
800 DEFUSR0=&HA010:X=USR0(Y)
810 COLOR 7:LOCATE6,0:PRINT"ｱﾗﾀﾒﾃ ﾅﾗ  f・1 ｦ｡ ﾂｲｶﾅﾗ   f・2(ﾄﾞｯﾄ),f・3(ｷｬﾗ) ｦ｡ ﾔﾒﾅﾗ
 RET ｦ  ｵｽｺﾄ ":GOTO 830
820 COLOR4:LOCATE6,0:PRINT"ｵｼﾁｶﾞｲ!  f・1(ﾂｷﾞｦｶｸ) f・2(ﾄﾞ=ﾂｲｶ) f・3(ｷｬ=ﾂｲｶ) RET(ｵﾜﾘ)
 ｦ  ｵｼﾅｵｽｺﾄ   ":COLOR7
830 Z$=INKEY$:IF Z$=""THEN 830
840 LOCATE6,0:PRINT"                             ｶｲﾒ ﾉ ﾂｲｶ｡     ｶｷｵﾜﾚﾊﾞ  f・1  ｦ  ｵｼﾃ
 ｸﾀﾞｻｲ":LOCATE28,0:PRINTQ
850 E=INP(9):R=INP(1)
860 IF R=127 THEN 870 ELSE IF E=247 THEN Q=Q+1:GOTO 460 ELSE IF E=251 THEN Q=Q+1
:GOTO 910 ELSE IF E=253 THEN 250 ELSE 820
870 PRINTCHR$(12)
```

```
880 COLOR7:LOCATE30,10:PRINT"ｶ ｷ ｺ ﾐ ｵ ﾜ ﾘ":LOCATE16,20:PRINT"ｶﾞｿﾞｰ ｻｲｾｲ ﾜ  f･8(
SHIFT+f･3)  ﾃｽﾄ ﾜ f･9(SHIFT+f･4) ﾃﾞｽ":LOCATE0,0:END

890 K=K+1:PRINTCHR$(12):COLOR 4:LOCATE26,5:PRINT"ﾒ ﾓ ﾘ  ｵ ｰ ﾊﾞ ｰ  ﾃﾞ ｽ":GOTO 880

900 PRINTCHR$(12)

910 D=7:X=40:Y=13

920 C=INP(6):GOSUB 670

930 S=INP(0):R=INP(1):GOSUB 570

940 E=INP(9):IF E=253 THEN 770

950 IF E=191 THEN 990

960 IF R=127 THEN 870

970 F=INP(8):IF F=191 THEN 1000

980 IF POINT(X,Y)=-1 THEN 920 ELSE PSET(X,Y):GOTO 920

990 IF POINT(X,Y)=-1 THEN PSET(X,Y):PRESET(X,Y):GOTO 920

1000 IF POINT(X,Y)=-1 THEN PSET(X,Y) ELSE PSET(X,Y):PRESET(X,Y)

1010 GOTO 920

1020 E$=INKEY$:F$=INKEY$:G$=INKEY$

1030 X=33:Y=0:Z=0

1040 D$=INKEY$:IF D$=""THEN 1040

1050 E=INP(9):IF E=253 THEN 1080

1060 LOCATEX,Y:PRINT D$

1070 X=X+1:Z=Z+1:IF Z=>8 THEN 1030 ELSE 1040

1080 DEFUSR3=&HA280:X=USR3(Y)

1090 RETURN
```

画像再生用プログラム・リスト

```
1100 PRINTCHR$(12):LOCATE22,10:COLOR4:PRINT"!!! ｶﾞﾒﾝ ｽｳ  ｵ ｰ ﾊﾞ ｰ !!! ｵｼﾅｵｼ!!!":
FOR I=1 TO 1500:NEXT I:COLOR7

1110 WIDTH,20:PRINTCHR$(12):LOCATE22,0:PRINT"* * ﾘ   ｽ   ﾄ * *  (ｷﾎﾞｰﾉ ｽｳｼﾞ ﾂｯﾞ
ｲﾃ  RET ｦ  ｵｽｺﾄ)":T=1:W=1:N1=500

1120 DEFUSR4=&HA2B0:X=USR4(Y):GOTO 1130

1130 LOCATE22,18:PRINT"ｷﾎﾞｰ ﾉ  ｶﾞﾒ ﾝ  ﾜ",

1140 LOCATE40,18:INPUT O

1150 LOCATE22,19:PRINT"ｿﾉ ｴ ｶﾗﾉ  ｶﾞﾒﾝｽｩ (ﾊﾐﾃﾞﾇｺﾄ)",

1160 LOCATE50,19:INPUT T

1170 WIDTH,25:O1=O*2

1180 G=PEEK(&H9DAE+O1):H=PEEK(&H9DAF+O1)

1190 POKE &HA000,G:POKE &HA001,H:GOTO 1210

1200 CONSOLE,,0,1:WIDTH80,25:POKE &HA000,&H30:POKE &HA001,&HA3:W=1:O=1:N1=1000:T
=PEEK(&H9FF2):GOTO 1230
```

●視力検査プログラム・リスト

```
1210 T2=PEEK(&H9FF7)/16+1:T3=(PEEK(&H9FF8)-&H9E)*16:T1=T2+T3

1220 IF (O+T)>T1 THEN 1100

1230 FOR J=1 TO T

1240 PRINTCHR$(12):DEFUSR1=&HA1D0:X=USR1(Y)

1250 O=O-1:W=16*O:O=O+2

1260 POKE &H9FF6,&H9F

1270 IF W<256 THEN POKE &H9FF6,&H9E ELSE W=W-256

1280 POKE &H9FF5,W

1290 DEFUSR5=&HA310:X=USR5(Y)

1300 FOR N=1 TO  N1:NEXT N

1310 NEXT J:LOCATE20,23:PRINT"ｽﾐ ﾏｼﾀ｡ ｵﾜﾘ ﾅﾗ  RETURN.  ﾂｲｶ ﾅﾗ  f･1 ｦ ｵｼﾃｸﾀﾞｻｲ"

1320 L=PEEK(&HA001):M=PEEK(&HA000)

1330 X$=INKEY$:IF X$="" THEN 1330

1340 E=INP(9):IF E=253 THEN 1370

1350 R=INP(1):IF R<>127 THEN 1330

1360 PRINTCHR$(12):LOCATE18,0:PRINT "ﾂｷﾞ ﾉ ｶﾞﾒﾝ No."O:LOCATE40,0:PRINT"ﾂｷﾞ ﾉ ｱﾄﾞﾚ
ｽ --"HEX$(L)HEX$(M):LOCATE24,12:PRINT"* *  ｺ ﾚ ﾃﾞ ｵ ｼ ﾏ ｲ ﾃﾞ ｽ  * * ":LOCATE0,0:
END

1370 LOCATE20,0:PRINT"                                          ":LOCATE2,0:PRINTHEX$(L)
HEX$(M):POKE &HA00D,M:POKE &HA00E,L:LOCATE20,23:PRINT"ｺﾚｶﾞ ｷｴﾀﾗ  f･2 (ﾄﾞｯﾄ) ﾏﾀﾜ
 f･3 (ｷｬﾗ) ｵ ｵｼﾃｸﾀﾞｻｲ"

1380 FOR P=1 TO 2000:NEXT P:LOCATE2,0:PRINT"      ":LOCATE20,23:PRINT"


1390 Z$=INKEY$:IF Z$="" THEN 1390

1400 E=INP(9):IF E=247 THEN 460

1410 IF E=251 THEN 910

1420 GOTO 1390
```

マシン語リスト

(1)画像ストア・プログラム（書き込み用）USR0 & HA010

```
A010 21 00 00 39 22 82 E9 31 80 E9 2A 0D A0 22 00 A0 :1A
A020 C5 D5 E5 F5 AF F5 06 15 21 78 F3 D9 00 00 00 00 :98
A030 21 C8 F3 11 78 F3 D9 0E 50 3E 00 BE 00 00 00 00 :8B
A040 C2 C0 A0 23 0D C2 3B A0 11 28 00 19 F1 3C F5 D9 :3C
A050 01 78 00 09 EB 09 EB D9 05 C2 37 A0 F1 00 00 00 :C9
A060 3A 0C A0 21 09 A0 96 3C 32 06 A0 3A 0B A0 00 00 :3F
A070 21 0A A0 96 32 07 A0 2A 00 A0 EB 21 00 A0 00 00 :B0
A080 01 10 00 ED B0 2A 02 A0 E5 CD 88 A1 2A 04 A0 00 :23
A090 CD 90 A1 CD 98 A1 05 C2 90 A0 E1 CD 88 A1 00 00 :D2
A0A0 CD B4 A1 CD 90 A1 CD 98 A1 05 C2 A3 A0 CD B4 A1 :52
A0B0 EB 22 0D A0 CD F8 A2 F1 E1 D1 C1 2A 82 E9 F9 C9 :DC
A0C0 F1 32 09 A0 D9,22 02 A0 EB 22 04 A0 D9 00 00 00 :F3
A0D0 21 78 F3 E5 21 78 F3 06 50 11 78 00 0E 15 3E 00 :3D
A0E0 BE C2 F5 A0 19 0D C2 E0 A0 E1 23 E5 05 C2 DC A0 :A9
A0F0 C3 60 A0 00 00 2A 04 A0 AF 3E 4F 90 4F 85 6F 00 :A0
A100 3E 00 8C 67 22 04 A0 21 0A A0 71 21 C7 F3 E5 00 :F3
A110 06 50 11 78 00 0E 15 3E 00 BE C2 30 A1 19 0D 00 :B7
A120 C2 19 A1 E1 2B E5 05 C2 15 A1 C3 60 A0 00 00 00 :AD
A130 21 0B A0 70 21 27 FD 06 15 3E 00 11 D8 FF 00 00 :C2
A140 0E 50 BE C2 54 A1 2B 0D C2 42 A1 19 05 C2 40 A1 :71
A150 C3 60 A0 00 21 0C A0 70 21 EE F3 E5 06 28 11 78 :9E
A160 00 00 00 00 0E 15 7E 19 BE C2 80 A1 0D C2 66 A1 :31
A170 E1 2B E5 05 C2 64 A1 C3 60 A0 00 00 00 00 00 00 :80
```

```
A180 21 08 A0 70 C3 60 A0 00 E5 21 06 A0 46 E1 C9 00 :98
A190 E5 21 07 A0 4E E1 C9 00 7E 12 23 13 0D 20 F9 CD :5E
A1A0 A4 A1 C9 00 D5 11 78 00 CD 90 A1 7B 91 5F 19 D1 :BF
A1B0 C9 00 00 00 D5 E5 2A 07 A0 55 5C EB 22 07 A0 E1 :9A
A1C0 D1 C9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :9A
A2F0 2A 8A E9 F9 C9 00 00 00 2A F3 9F 01 00 A0 0A 77 :3D
A300 03 23 0A 77 23 22 F3 9F C9 00 00 00 00 00 00 00 :47
```

```
A1D0 21 00 00 39 22 84 E9 31 80 E9 2A 00 A0 11 00 A0 :FE
A1E0 01 10 00 ED B0 C5 D5 E5 F5 2A 00 A0 EB 21 10 00 :08
A1F0 19 EB 2A 02 A0 E5 CD 88 A1 2A 04 A0 CD 90 A1 CD :44
A200 30 A2 05 C2 FC A1 E1 CD 88 A1 CD B4 A1 00 00 00 :2F
A210 CD 90 A1 CD 30 A2 05 C2 10 A2 CD B4 A1 EB 00 00 :23
A220 22 00 A0 F1 E1 D1 C1 2A 84 E9 F9 C9 00 00 00 00 :7F
A230 1A 77 23 13 0D 20 F9 CD A4 A1 C9 00 00 00 00 00 :C8
```

```
A240 21 00 00 39 22 86 E9 31 80 E9 C5 D5 E5 00 3E FF :41
A250 21 00 9E 06 02 0E FE 77 23 0D 20 FB 05 20 F6 21 :D1
A260 F0 9F 36 00 23 36 9E 23 36 00 23 36 B0 23 36 9D :14
A270 21 0D A0 36 30 23 36 A3 E1 D1 C1 2A 86 E9 F9 C9 :FE
```

```
A280 21 00 00 39 22 88 E9 31 80 E9 C5 D5 E5 F5 21 F0 :0C
A290 9F 5E 23 56 21 1E F3 01 10 00 ED B0 21 F0 9F 73 :79
A2A0 23 72 23 4E 0C 00 71 F1 E1 D1 C1 2A 88 E9 F9 C9 :44
```

```
A2B0 21 00 00 39 22 8A E9 31 80 E9 C5 D5 E5 F5 21 00 :1E
A2C0 9E 11 00 F4 06 02 0E 12 C5 01 10 00 ED B0 E5 21 :44
A2D0 68 00 19 EB E1 7E FE FF 00 CA E8 A2 C1 0D C2 C8 :74
A2E0 A2 11 20 F4 05 C2 C6 A2 22 F7 9F E1 F1 E1 D1 C1 :F3
A2F0 2A 8A E9 F9 C9 00 00 00 2A F3 9F 01 00 A0 0A 77 :3D
```

```
A310 21 00 00 39 22 8C E9 31 80 E9 C5 D5 E5 2A F5 9F :C8
A320 11 20 F3 01 10 00 ED B0 E1 D1 C1 2A 8C E9 F9 C9 :A6
```

テレビ・アニメーション編集プログラム・リスト

```
10 ' テ レ ビ ア ニ メ  ヘ ン シ ュ ー プ ロ グ ラ ム

20 CLEAR 500,&H9D9F:WIDTH80,25:CONSOLE,,0,1:KEY8,"Print":KEY9,"goto170"+CHR$(13)
:GOTO 40

30 PRINTCHR$(12):LOCATE22,10:COLOR4:PRINT"!!! ガメン スウ  オ ー バ ー !!! オシナオシ!!!":FO
R I=1 TO 1500:NEXT I:COLOR7

40 WIDTH,20:PRINTCHR$(12):LOCATE22,0:PRINT"* * リ   ス   ト * *   (キボーノ スウジ ツヅイテ
  RET ヲ  オスコト)":T=1:W=1:N1=500:T4=1

50 DEFUSR4=&HA2B0:X=USR4(Y)

60 LOCATE22,17:PRINT"キボー ノ  ガメ ン  ワ",

70 LOCATE40,17:INPUT O

80 LOCATE22,18:PRINT"ソノ エ カラノ  ガメンスゥ (ハミデヌコト)",

90 LOCATE50,18:INPUT T

100 LOCATE22,19:PRINT"ナンカイ  クリカエシ マスカ",

110 LOCATE48,19:INPUT T4
```

テレビ・アニメーション編集プログラム・リスト

```
120 WIDTH,25:O2=0:O1=O*2

130 FOR J1=1 TO T4

140 G=PEEK(&H9DAE+O1):H=PEEK(&H9DAF+O1)

150 POKE &HA000,G:POKE &HA001,H:GOTO 190

160 '55,5,20 by T.KONDO

170 CONSOLE,,0,1:WIDTH80,25:POKE &HA000,&H30:POKE &HA001,&HA3:W=1:O=1:N1=1000:T4
=1:T=PEEK(&H9FF2)

180 FOR J1=1 TO T4:GOTO 210

190 T2=PEEK(&H9FF7)/16+1:T3=(PEEK(&H9FF8)-&H9E)*16:T1=T2+T3

200 IF (O+T)>T1 THEN 30

210 FOR J=1 TO T

220 PRINTCHR$(12):DEFUSR1=&HA1D0:X=USR1(Y)

230 O=O-1:W=16*O:O=O+2

240 POKE &H9FF6,&H9F

250 IF W<256 THEN POKE &H9FF6,&H9E ELSE W=W-256

260 POKE &H9FF5,W

270 DEFUSR5=&HA310:X=USR5(Y)

280 FOR N=1 TO  N1:NEXT N

290 NEXT J:O3=O:O=O2:NEXT J1

300 L=PEEK(&HA001):M=PEEK(&HA000)

310 PRINTCHR$(12):LOCATE18,0:PRINT "ﾂｷﾞﾉ ｶﾞﾒﾝ No."O3:LOCATE40,0:PRINT"ﾂｷﾞ ﾉ ｱﾄﾞﾚ
ｽ --"HEX$(L)HEX$(M):LOCATE24,12:PRINT"* *  ｺ ﾚ ﾃﾞ ｵ ｼ ﾏ ｲ ﾃﾞ ｽ  * * ":LOCATE0,0:
END
```

ゲーム編





PC-8001 BASIC＋マシン語

清水邦敏　宇野　敦

地獄の黙示録

遊び方

画面中央下部に現われるヘリコプターを操縦しながら基地に爆弾を投下して破壊するのです．ヘリコプターはテンキーの[1]で降下，[4]で上昇し，[GRPH]キーで爆弾投下です．上空では空対空ミサイルが，地上からは地対空ミサイルが飛んでくるので避けてください．また，地面に衝突してもゲーム・オーバーです．なお，画面にはヘリコプターの飛行軌跡と爆弾の落下軌跡が表示されます．

得点について

手抜きをしたため，爆弾の投下した位置の地面からの高さに関係なく基地１つで１点です．高得点を得るには，地面スレスレを地面に沿って低空飛行し，[GRPH]キーを押しっぱなしにすればいいのです．これは爆弾が何かに当るまで次の爆弾が投下できないことを考えればわかると思います．

プログラム

❶BASICは表１に示すとおりです．

行番号	内　容
40～230行	初期設定
240行	地形を乱数で決める．
250行	基地の出現率を乱数で決める．
260行	地対空ミサイルを乱数により表示
270行	空対空ミサイルを乱数により表示
280行	左端に地形を表示
290行	マシン語ルーチン（横スクロール・キー処理など）
300行	左端の地形を消す．
320～390行	GAME OVERの処理

❷マシン語は簡単な命令しか使っていないので，逆アセンブルすればすぐわかると思います．ただし，D100～D11Fはワーク・エリアです．

変更点

BASICの説明を見ればわかるように，250～270行の乱数の比較値を変えれば良いのです．

その他

ミサイルがヘリに当たっても，ゲーム・オーバーにならないことがありますが，これはミサイルがヘリの横をそれたと思ってください．

このゲームは白黒です．カラー化することは横スクロールをやっていることを考えれば，アトリビュート操作が信じられないくらい，面倒になり，まず不可能でしょう．小生のようにグリーン・モニタしか持っていない人間には好都合でした．

なお，スコアには１バイトしか使っていないので，256点以上取ると０点に戻りますが，256点取るなど絶対に不可能でしょう．

プログラムを入れるときの注意ですが，マシン語を入れたら走らせる前に必ずセーブしておいてください．いくらリセット・スイッチがあっても，暴走してメモリの内容が変わってしまうと目も当てられません．

90～180行のグラフィック・キャラクタはコード番号&H95，&H96，&H9C～&H9Fです．基地はクラブのマークです．

参考文献

1）Z80マイコンプログラミングテクニック，電波新聞社

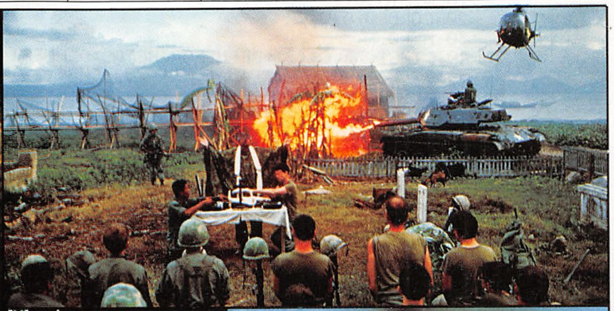
タイトル写真提供：日本ヘラルド映画

★この世ならぬ２時間半の旅は，あなたの
脳裏に永遠に残る！

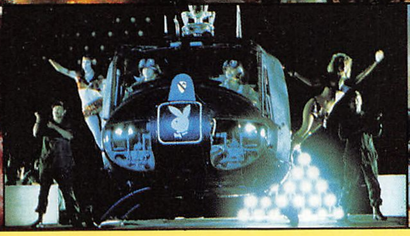
MOVIE

地獄の黙示録











■あなたは６人目の同乗者！

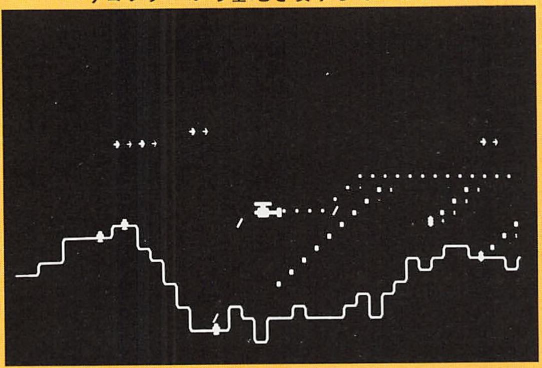
３そうのＰＢＲ(河川巡回艇)に乗って戦争をするひとりの大尉と４人の部下．彼らが目撃するすべてのことを，６人目の同乗者となって同時体験する２時間半．
サイゴンの情報指令部に呼び出されたウィラード大尉（マーチン・シーン）は，数々の戦歴で勇名をはせるカーツ大佐（マーロン・ブランド）の暗殺を命じられる．カーツ大佐はナン川上流のジャングル奥地に“王国”を築いて，支配者に収まっているというのだ．
ウィラードは，クリーン(ラリー・フィッシュバーン)ランス（サム・ボトムズ），シェフ（フレデリック・フォレスト），チーフ（アルバート・ホール）の４人を連れて，川をさかのぼった．
空軍騎兵隊第１中隊のキルゴア中尉（ロバート・デュバル）のもとでの凄まじい殺りく，次に出会った密林の中のラスベガス．平常の感覚を次第に失ってゆく彼らが遂にたどりついた最終目的地には，白塗りの不気味な男たちが待ちかまえていた．カーツは狂人なのか，それとも偉大な指導者なのか．フォト・ジャーナリスト（デニス・ホッパー）に案内されて，ウィラードはいま神の館に足を踏み入れた．

キャスト

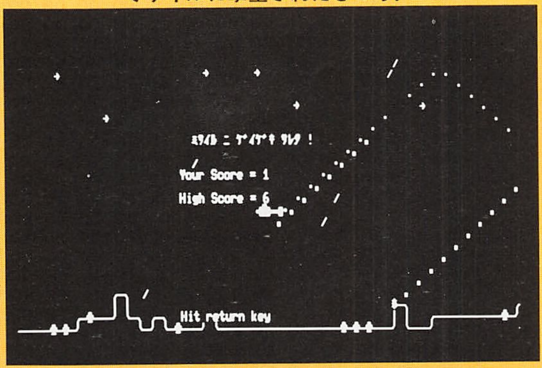
マーロン・ブランド……………………………カーツ大佐
ロバート・デュバル……………………………キルゴア中佐
マーチン・シーン………………………………ウィラード大尉
フレデリック・フォレスト………………………シェフ
アルバート・ホール……………………………チーフ
サム・ボトムズ…………………………………ランス
ラリー・フィッシュバーン………………………クリーン
デニス・ホッパー………………ニュース・カメラマン

写真・資料提供：日本ヘラルド映画

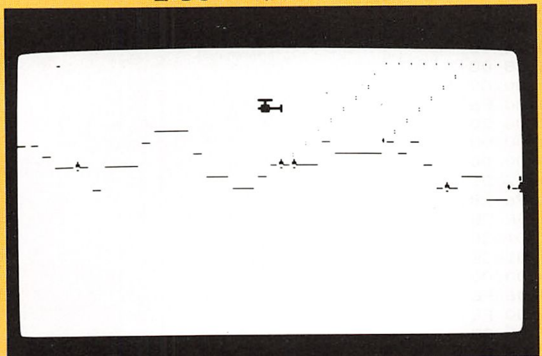
ヘリコプターから基地を攻撃しているところ．



ミサイルに撃墜されたところ．



基地を１つ爆破したところ．



地面に衝突したところ．



BASIC プログラム・リスト

```
10 WIDTH80,25:CONSOLE0,25,0
20 COLOR0,0,0
30 HS=0
40 DIMG$(7),C$(7):DEFUSR0=&HCFD8:DEFUSR1=&HD140
50 C$(2)="  "+CHR$(30)+CHR$(29)+CHR$(29)+"  "+CHR$(30)+CHR$(29)+CHR$(29)+"  "
60 G$(2)=" ＼"+CHR$(30)+CHR$(29)+CHR$(29)+" |"+CHR$(30)+CHR$(29)+CHR$(29)+"─¬"
70 C$(3)="  "+CHR$(30)+CHR$(29)+CHR$(29)+"  "
80 G$(3)=" ＼"+CHR$(30)+CHR$(29)+CHR$(29)+"─¬"
90 G$(4)="──":B$="¬*"
100 S$="──"
110 C$(4)="  "
120 G$(5)=" ╱"+CHR$(31)+CHR$(29)+CHR$(29)+"─╯"
130 C$(5)="  "+CHR$(31)+CHR$(29)+CHR$(29)+"  "
140 C$(6)="  "+CHR$(31)+CHR$(29)+CHR$(29)+"  "+CHR$(31)+CHR$(29)+CHR$(29)+"  "
150 G$(6)=" ╱"+CHR$(31)+CHR$(29)+CHR$(29)+" |"+CHR$(31)+CHR$(29)+CHR$(29)+"─╯"
160 DATA &H88,&H88,&H08,0,&He4,&Hef,&H44,&He4,0,0,0,0,0,0,0,0,&H76,&Hfd,&Hee,&Hf
d,&Ha3,&Hfd,0,0,0,0,0,&Hf4
170 FORI=&HD100TO&HD11B:READ A:POKEI,A:NEXT
180 PRINT CHR$(12)
190 FORI=&HF350TO&HFE90STEP120:POKEI,0:POKEI+1,0:POKEI+2,38:POKEI+3,0:POKEI+4,42
:POKEI+5,0:NEXT:POKE&HD11E,1
200 Y=24:LOCATE2,24:PRINT STRING$(77,&H95);
210 G1=INT(RND(1)*7+1):G2=Y-4+G1:IF G2<5OR G2>23ORG1=1ORG1=7THENG1=4:G2=Y
220 IFG1=4ANDRND(1)<.3THENG$(G1)=B$
230 IFRND(1)<.3THENPOKE&HD11A,INT(RND(1)*37+&H6A):A=USR1(0)
240 IFRND(1)<.1THENPOKE&HF37A+120*INT(RND(1)*5),28
250 LOCATE0,Y:PRINTG$(G1);:G$(4)=S$
260 A=USR0(0):IF PEEK(&HD116)<>0THEN290
270 LOCATE0,Y:PRINTC$(G1);:Y=G2
280 GOTO 210
290 A=PEEK(&HD116):LOCATE28,7:IFA=2THENPRINT "ｼﾞﾒﾝ ﾆ ｼｮｳﾄﾂ !";ELSEPRINT "ﾐｻｲﾙ ﾆ
ｹﾞｲｹﾞｷ ｻﾚﾀ !";
300 S=PEEK(&HD117)
310 LOCATE26,10:PRINT "Your Score =";S
320 LOCATE26,12:PRINT "High Score =";HS
330 IF S>HS THENHS=S:LOCATE26,14:PRINT "You made High Score"
340 LOCATE26,22:PRINT "Hit return key";:BEEP1:OUT81,33:FORI=1TO600:NEXT:OUT81,32
:BEEP0
350 IF INKEY$<>CHR$(13)THEN350 ELSE RESTORE:GOTO 170
360 END
```

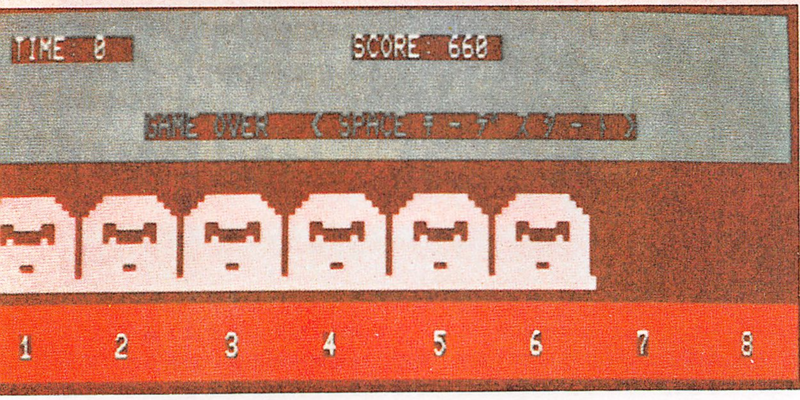
マシン語プログラム

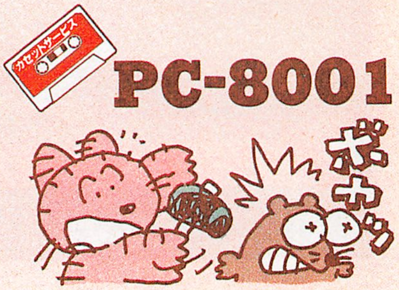
```
CFB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
CFC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
CFD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
CFE0  2A 14 D1 11 78 00 3E 00 77 19 77 21 08 D1 ED 5B
CFF0  10 D1 01 04 00 ED B0 01 04 00 ED 5B 12 D1 ED B0
D000  26 2F 11 C6 00 0E 19 DD 21 D5 F2 06 4E DD 7E 78
D010  BC 20 05 DD 77 02 3E 20 DD 77 7A DD 2B 10 EE DD
D020  19 0D 20 E7 DB 00 06 FD 0E EF B8 28 0D B9 28 05
D030  11 00 00 18 08 11 88 FF 18 03 11 78 00 2A 14 D1
D040  19 3E DB BD 20 03 21 53 F3 3E 80 77 22 14 D1 01
D050  78 00 09 77 2A 10 D1 19 3E AE BD 20 03 21 26 F3
D060  22 10 D1 2A 12 D1 19 3E 26 BD 20 03 21 9E F3 22
D070  12 D1 CD A0 D0 21 00 D1 ED 5B 10 D1 01 04 00 ED
D080  B0 ED 5B 12 D1 01 04 00 ED B0 2A 12 D1 23 23 23
D090  23 3E A5 77 CD B0 D1 C9 00 00 00 00 00 00 00 00
D0A0  06 04 2A 12 D1 7E FE 2F 28 1E FE 9C 28 1E FE 9D
D0B0  28 1A FE 9E 28 16 FE 9F 28 12 FE 95 28 0E FE 96
D0C0  28 0A FE 1C 28 02 18 0A 3E 01 18 02 3E 02 11 16
D0D0  D1 12 23 10 D0 C9 21 20 D1 11 C4 F7 01 1A 00 ED
D0E0  B0 AF 32 1E D1 C9 00 00 00 00 00 00 00 00 00 00
D0F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D100  88 88 08 00 E4 EF 44 E4 00 00 00 00 00 00 00 00
D110  86 FC FE FC B3 FC 02 03 00 00 73 F4 89 FC 00 00
D120  54 4F 48 4F 4B 55 20 55 4E 49 56 20 20 55 4E 4F
D130  20 26 20 53 48 49 4D 49 5A 55 FF FF FF FF FF FF
D140  06 2F 11 78 00 DD 2A 1A D1 DD 7E 78 FE 95 28 1C
D150  FE 96 28 18 FE 9C 28 14 FE 9D 28 10 FE 9E 28 0C
D160  FE 9F 28 08 FE EB 28 04 DD 19 18 DD DD 70 00 C9
D170  3E 21 D3 51 0E 70 3E FF 3D FE 00 20 FB 0D 20 F6
D180  3E 20 D3 51 3E 20 D3 40 0E 10 3E FF 3D FE 00 20
D190  FB 0D 20 F6 3E 00 D3 40 C9 00 00 00 00 00 00 00
D1A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D1B0  3A 18 D1 FE 00 C2 10 D2 DB 08 FE EF C0 2A 12 D1
D1C0  11 78 00 19 7E FE EB 28 2C FE 95 28 32 FE 96 28
D1D0  2E FE 9C 28 2A FE 9D 28 26 FE 9E 28 22 FE 9F 28
D1E0  1E FE 2F 28 1A FE 2A 28 16 3E A1 77 22 1C D1 3E
D1F0  01 32 18 D1 C9 3A 17 D1 3C 32 17 D1 CD 70 D1 3E
D200  2A 77 AF 32 18 D1 C9 00 00 00 00 00 00 00 00 00
D210  2A 1C D1 11 78 00 01 76 00 19 7E FE EB 28 D6 FE
D220  95 28 DC FE 96 28 D8 FE 9C 28 D4 FE 9D 28 D0 FE
D230  9E 28 CC FE 9F 28 C8 FE 2F 28 C4 09 7E FE 2F 28
D240  0E FE 1C 28 0A A7 ED 42 3E A1 77 22 1C D1 C9 AF
D250  77 32 18 D1 3E 2A A7 ED 42 77 C9 00 00 00 00 00
```

チェック・サム

```
CFB0 - CFEF = 19A0
CFF0 - D02F = 19B0
D030 - D06F = 1395
D070 - D0AF = 1837
D0B0 - D0EF = 1386
D0F0 - D12F = 0F60
D130 - D16F = 1E13
D170 - D1AF = 1098
D1B0 - D1EF = 1CAF
D1F0 - D22F = 1AC4
D230 - D25F = 1431
```

モグラたたきゲーム





PC-8001

徳田　進

本誌1980年３月号島田啓一郎氏作の『スズメたたきゲーム』を見て，PC-8001で『モグラたたきゲーム』をと思い立ちました．BASICなのでスピードの点で若干不安でしたがPC-8001の強力なカラーグラフィックス機能を生かして作ったゲームです．

このゲームは色を楽しんでください．高解像度カラーディスプレで楽しめば，面白さも倍加するはずです．



ルール

ゲームセンターなどにあるモグラたたきゲームと同様で，このゲームのルールはモグラがランダムに１～８の番号上に現われたり，消えたりするので，モグラが現われたら素早くその番号のキーを押し，一定時間に数多くモグラを叩けばよいという簡単なルールです．



遊び方

まず，プログラム・リストをすべて打ち込み，ＲＵＮさせると**写真１**のように初期表示としてモグラが８匹，次々と画面に表示されます．

次に，**スペース**キーを押すとゲーム開始で持ちTIME：60，SCORE：100が表示され，モグラがランダムに１～８の番号上に現われます(**写真２**)．そのとき素早くその番号のキーを押すと，モグラが叩かれて『ピッピッピッ』と鳴いて（実際にモグラは鳴くか小生は知りませんが…？），モグラが紫色になり恐って（恐ったつもりです）消えます(**写真３**)．そのとき，得点が20点加算されます．失敗すると10点減点されます．

TIME：０またはSCORE：０になるとゲーム・オーバー

筆者のシステム全景



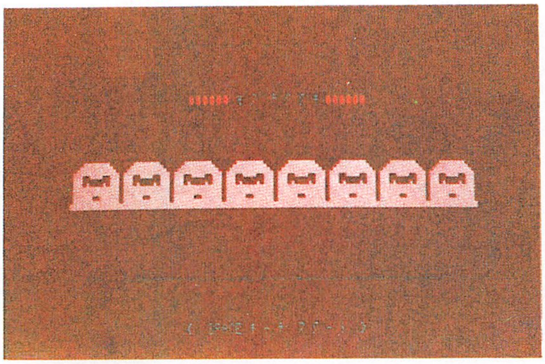


写真１　初期の表示(タイトル)

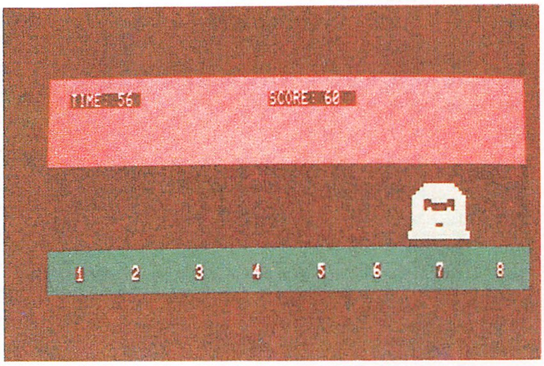


写真２　モグラがランダムに動き廻み

写真3　モグラが叩かれて紫色に変わった

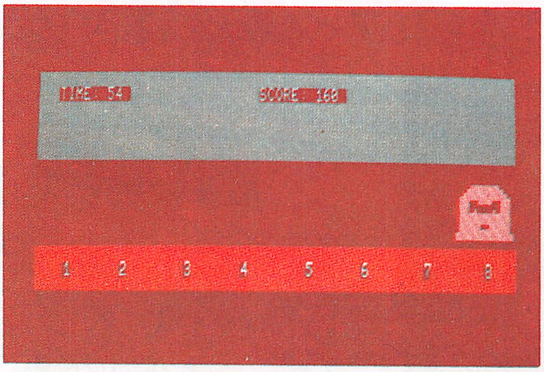
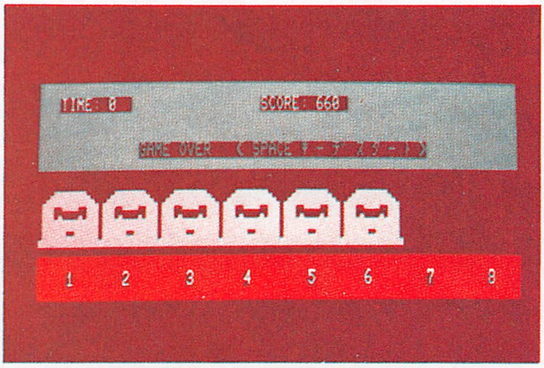


写真4　ゲーム終了時の表示



で，GAME OVERとHIGH SCOREが表示されて，1ゲームが終了します(写真4)．時間の表示は最初，持ち時間として60が表示され，1ずつ減っていきます．この－1される時間は実際の1秒ではなく，単なるゲームのための時計用となっています．このことについては，プログラムの方で触れることにします．

なお，ゲームの最高得点は，1,300点にしてあります．



プログラム

プログラムは，すべてBASICで書いてあります．短いプログラムなので，表1に示したプログラムのアドレス・マップを見ていただければ，説明するほどのことはないと思います．

ただ，プログラムの中でモグラ，地面などの表示には，N-BASICのグラフィックス命令PSET，PRESET，PUT＠，LINEなどを使っています．これらの命令は，BASICとしてはかなりのスピードでグラフィックス・パターンなどの移動，表示，消去などが簡単にできるので大変便利だと思います．

遊び方の項でちょっと触れましたが，時間の表示はモグラが表示され次に新しいモグラを表示するまでの処理時間(実際は約1.6秒)で時間の表示を－1しています．一方，モグラが叩かれるとサブルーチンに入りサブルーチンを通った分だけ処理時間(この場合は約2.5秒要する)が長くなりますが，このときも同様に時間の表示は1減らすようになっています．実際には処理時間の違いがあるわけですが，ゲームにおいてはさほど不自然さは感じられないと思います．

表1　アドレス・マップ

行番号	内　容
10～120	初期表示(50～70モグラのデータ読み込み)
130～140	スペース キーが押されたか検索
160～250	地面および番号の表示
280	TIME，SCOREの初期値セット
285～300	TIME，SCOREの表示
330～370	モグラをランダムに表示および音出し
380～540	番号キーのセンス
560	モグラを消す
570	TIME，SCOREの値を処理
580	時間待ち(ダミー)
600～710	ゲーム終了時の画面表示処理
720～770	再ゲーム・スタート処理
780～830	サブルーチン(モグラの色，鳴かせる，SCOREの処理)
850～870	モグラのデータ

ゲームのスピードを変えるには，350行目を変えれば変化します．350行のFor　Q＝0　To 2を，For　Q＝0　To1とすれば，スピードは2倍になりかなり難かしくなります．スピードの点で物足りない方は試してみてください．逆にスピードを遅くすれば，幼児用のスピードとなります．

ゲームをする人が，好みのスピードを選択できるようにすることもINKEY$あるいはINPUTなどを使えば簡単にできると思いますのでプログラムを作ってみてください．

このプログラムはカラー用に作っていますが，グリーン・モニタなどで表示させる場合は輝度を上げるか，プログラム中のCOLOR命令の部分を変えれば結構楽しめると思います．



むすび

当初，懸念していたスピードの点もN-BASICの強力なカラーグラフィックス機能に助けられてできたゲームだと思っています．やはり，よりスピードを要求される部分は機械語で作る必要があると思います．

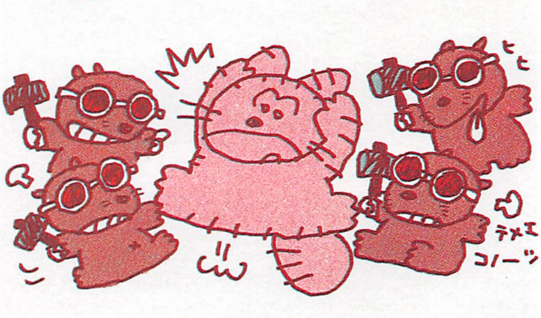
◘参考文献

1）島田啓一郎：“スズメたたきゲーム”，I/O，1980年3月号

2）NEC：PC-8001ユーザーズ・マニュアル，N-BASICレファレンス・マニュアル，1979年9月

3）後藤富雄：“グラフィックス機能とその活用法”，インターフェース，1979年12月号，CQ出版

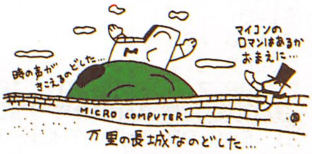
4）飯田・今井・大野・土田：N-BASIC入門，アスキー出版



```
10  REM******** Mogura *********
20  WIDTH 80,25:CONSOLE 0,25,0,1
30  PRINT CHR$(12):COLOR 2,0,0
40  LOCATE 25, 4:PRINT"               "
50 COLOR 5,0,0:LOCATE 31,4:PRINT" モ ク゛ ラ タ タ キ"
60  DIM A%(16)
70  FOR I=0 TO 16:READ A%(I):NEXT
80 COLOR 6,0,1
90 FOR M=0 TO 105 STEP 15
100 PUT@(15+M,35)-(30+M,50),A%,PSET:BEEP     :NEXT
110 COLOR 1,0,1:LINE(9,0)-(140,70),PSET,B
120 COLOR 5,0,0:LOCATE 24,21:PRINT" <  SPACE キ - テ゛ ス タ - ト  >"
130 A$=INKEY$:IF A$="" THEN 130
140 IF A$<>CHR$(32) THEN 130
150 BEEP:PRINT CHR$(12)
160 COLOR 2,0,1 :LINE(15,52)-(136,62),PSET,BF
170 COLOR 7,0,0
180 LOCATE 11,14:PRINT"1"
190 LOCATE 18,14:PRINT"2"
200 LOCATE 26,14:PRINT"3"
210 LOCATE 33,14:PRINT"4"
220 LOCATE 41,14:PRINT"5"
230 LOCATE 48,14:PRINT"6"
240 LOCATE 56,14:PRINT"7"
250 LOCATE 64,14:PRINT"8"
260 REM*******************************
270 S1=0
280 S=100 :T= 60
285 COLOR 5,0,1:LINE(15, 8)-(135,30),PSET,BF
290 COLOR 7,0,0:LOCATE 10,3:PRINT "TIME:";   T
300 LOCATE 35,3:PRINT "SCORE:";   S
310 IF S=0 OR T=0  THEN GOTO 600
320 COLOR 6,0,1
330 A=INT(RND(3)*10):IF A>8 THEN 330
340 X=A*15:IF X<15 THEN 330
350 FOR Q=0 TO 2
360 PUT@(X,35)-(X+15,50),A%,PSET:IF Q=0 THEN BEEP 1
370 FOR J=0 TO 13 :NEXT:BEEP 0
380 IF X=15 THEN 470
390 IF X=30 THEN 480
400 IF X=45 THEN 490
410 IF X=60 THEN 500
420 IF X=75 THEN 510
430 IF X=90 THEN 520
440 IF X=105 THEN 530
450 IF X=120 THEN 540
460 IF X=135 THEN 730
470 A$=INKEY$:IF A$=CHR$(49) THEN GOSUB 780
480 A$=INKEY$:IF A$=CHR$(50) THEN GOSUB 780
490 A$=INKEY$:IF A$=CHR$(51) THEN GOSUB 780
500 A$=INKEY$:IF A$=CHR$(52) THEN GOSUB 780
510 A$=INKEY$:IF A$=CHR$(53) THEN GOSUB 780
520 A$=INKEY$:IF A$=CHR$(54) THEN GOSUB 780
530 A$=INKEY$:IF A$=CHR$(55) THEN GOSUB 780
540 A$=INKEY$:IF A$=CHR$(56) THEN GOSUB 780
550 NEXT Q
560 PUT@(X,35)-(X+15,50),A%,PRESET
570 S=S-10 :T=T-1
580 FOR L=0 TO  700:NEXT
590 GOTO 290
600 COLOR 5,0,0: LOCATE 20,6:PRINT"GAME OVER   < SPACE キ - テ゛ ス タ - ト >"
610 IF S1>S THEN 630
620 S1=S
630 FOR R=0 TO 8
640 FOR U=0 TO 30:NEXT U: LOCATE 24,20:PRINT"                                 ":BE
EP 1
650 LOCATE 24,20:PRINT"TODAY'S  HIGH  SCORE:";    S1:BEEP 0:NEXT R
```

モグラたたきゲーム　プログラム・リスト

```
660 COLOR 6,0,1
670 FOR M=0 TO 105 STEP 15
680 PUT@(15+M,35)-(30+M,50),A%,PSET
690 A$=INKEY$:IF A$="" THEN 700 ELSE 720
700 NEXT
710 LINE(135,50)-(15,35),PRESET,BF    :GOTO 670
720 IF A$<>CHR$(32) THEN 690
730 BEEP:LINE(15,35)-(135,50),PRESET,BF
740 LOCATE 20,6:PRINT "
750 LOCATE 24,20:PRINT"
760 FOR Z=0 TO 700 :NEXT
770 GOTO 280
780 REM******* SUB ********************
790 COLOR 3,0,1: S=S+30
800 FOR G=0 TO 45
810 BEEP 1:FOR M=0 TO 8 :NEXT
820 BEEP 0:NEXT G
830 RETURN
850 DATA 256,      0, 4080, 8184,16380,32766,29646,28686
860 DATA 29742,32766,32766,32766,32382,32766,    -1,   -1
870 DATA     0,     0,
```



カセットサービス

PC-8001 BASIC

FIRE RESCUE

ファイア・レスキュー



キミは
炎上するビルから
何人救出できるか!?

1人の放火魔がビルの1階に火をつけた．火の勢いは劣えをみせず，序々に上へ向けて火の手を延ばしていく．各階では救出を待つ人が見えるが，救出できる手段はハシゴ車を使って1名ずつ避難させるのがやっと．

レスキュー隊長に扮するキミの使命は沈着な判断で，いかに多くの人を助けるかにある．時折くる消火用ヘリコプターが唯一の救いだ！

中台　進

ゲームの説明

ＲＵＮさせると最初に車が現われ，次にゲームの説明とデモが現われます．このゲームは，車を移動させハシゴを上下させて人を救助するゲームで，動きは画面に示されているとおり，[4]，[6]，[SHIFT] で動きます．

ハシゴの位置を間違えて1キャラクタだけ上にすると，人が押しつぶされて減点となります．減点はKILLと書いてあるところがそうです．

ハシゴの位置がぴたり合うとＳＡＦＥだけ加点，間に合わないで人が落ちるとＯＵＴの減点となります（黄色の人は，落ちても減点がありません）．

1番上の紫色の人は端にいるときしか救助できませんが，100POINTの加算となります．紫の人は落ちることはありません．

火が付けられてから上にあがりきるまでに400POINTを得点するとヘリコプターが飛んできて消火にあたります．火は全部は消えません．

GAME STARTは[−]キーで行ないますが，デモの場合，人が走っているとき，車が動いているとき（人を乗せないで)，ハシゴが上下しているとき，火がビルを包んだときに有効です．GAME START後にGAME STARTに戻したいときには，ハシゴが上下しているときに[−]キーを押してください．GAME OVERではデモに戻るので，続けたいときにはGAME OVERの状態のときに[−]キーを押してもGAME STARTに戻ります．

プログラムはN-ＢＡＳＩＣで書かれています．各ルーチンごとの説明を**表1**に示します．

注　[−]はマイナス・キーです．

変更点

このゲームでは人の落ちる時間がヘリコプターが飛ぶごとに速くなっていきますが，この時間の初期値は文番号，460のＺ＝34を多くすると遅く，少なくすると速くなります．火の動く速度は文番号620のTIME＝＞"00：00：02" を変えてください．上の紫色の人の動きは文番号540のE＞13を変えてください．多くすると遅く，少なくすると速くなります．

ヘリコプターのPOINTは文番号610のP＞390を変えてください．このゲームは現実性と見ため（車ということがよくわからなくなる）を考えて少し遅く作ってありますが，

写真１　まずは，ゲームの説明から…

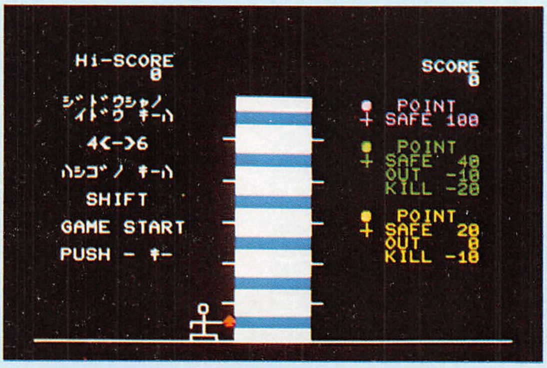


写真２　さっそうと(？)，左から放火魔が登場

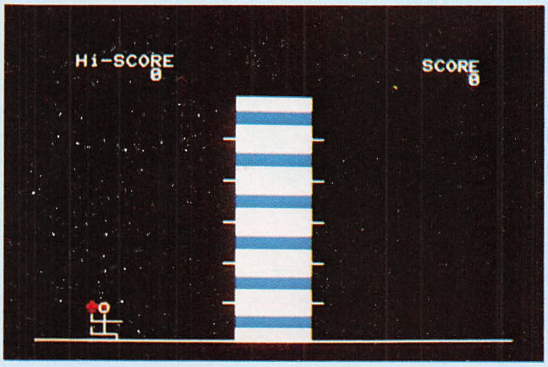


写真３　ハシゴ車で１人を無事救出

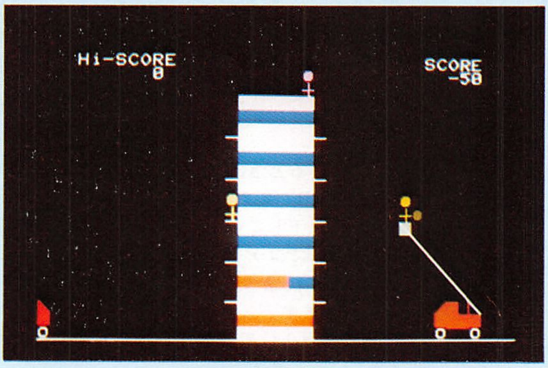


写真４　ハシゴ車が間に合わないと，飛び降る人が続出

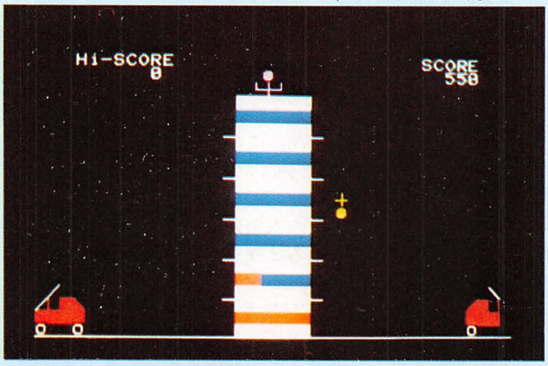


写真５　ハシゴの位置が合わないと死者が出て，減点

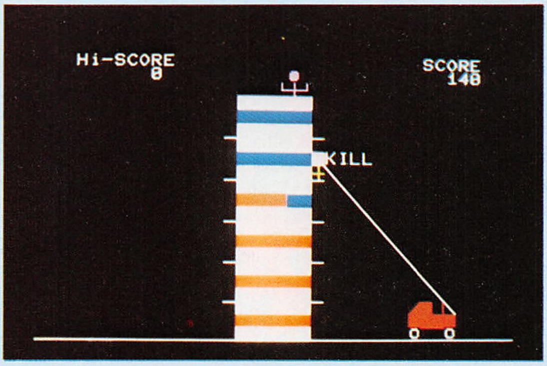


写真６　最上階の人を救け出すと，100点が加算される．

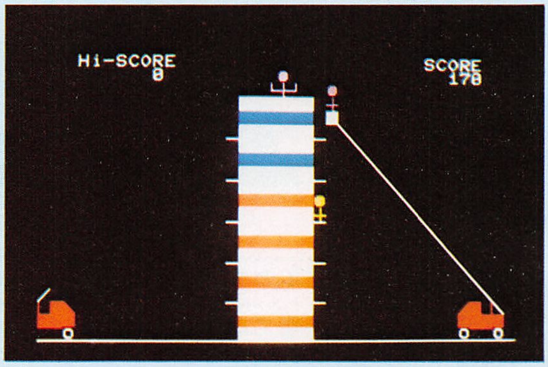


写真７　火の手が回りビルが全焼すると，ゲーム・オーバー

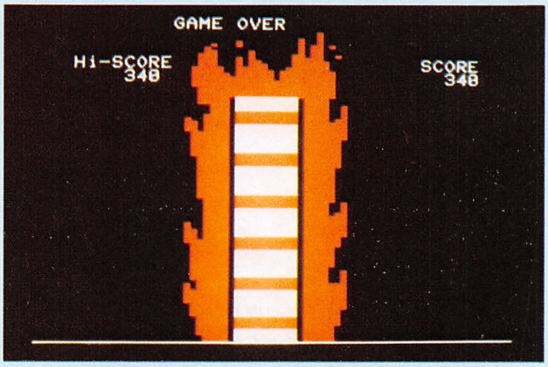


写真８　400点に達すると，消火用ヘリコプターが来る．

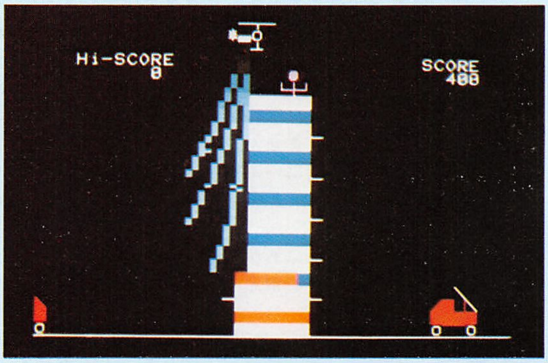
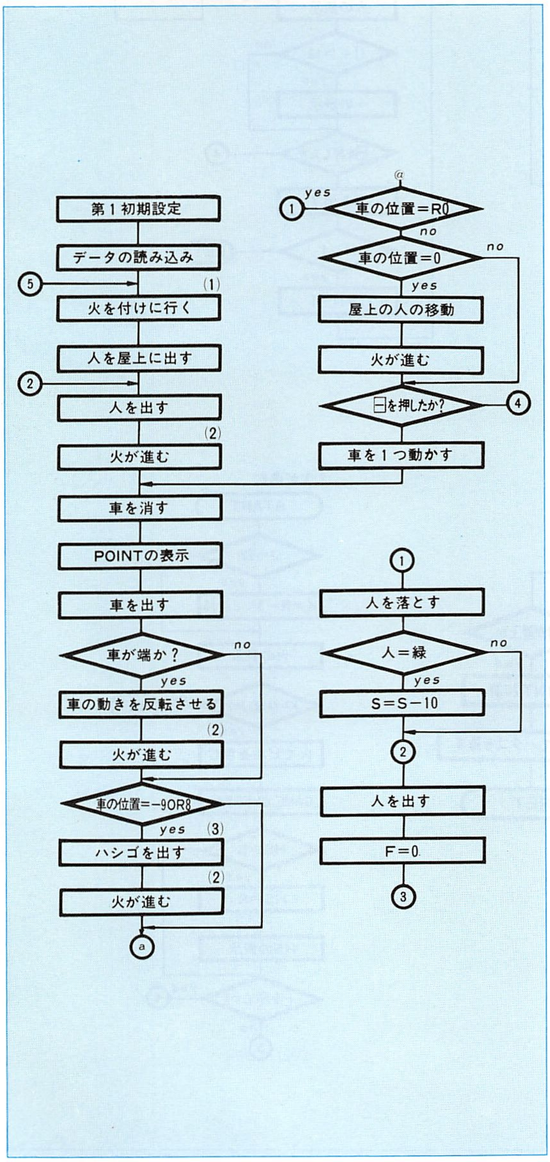


表1　プログラムの説明

行番号	ルーチン名	内容
50～ 110	DATA READ	ビルを包む炎，ヘリコプターの消火液をDATAから読み込み，車を表示してGETします．
120～ 330	FIRST SCENE	プログラムの説明やHI-SCOREの表示を与えます．
340～ 440	DEMO	人を走らせた後で車，火などの初期値を与えます．デモを繰り返して行ないます．
450～ 620	GAME	4，6，SHIFTキーを使って動きを行ないます．ハシゴ1回の上下を車の動きの3個かつ5個分として表わします．得点400POINTでヘリコプターのサブルーチンに飛ばします．
640～ 670	CAR PUT OUT	車の位置にあるものをすべて消します．
680～ 710	TOP MAN FIRST POSITION	最初の紫色の人を表示します．紫色の人が端で助けられたときにもこの位置に表示します．
720～ 880	CAR PUT	ON GOTOで位置に合わせた表示を行ないます．
890～ 970	BILL	ビルとベランダの表示をします．
980～1020	FIRE UP	火を1つずつ動かしていきGAME OVERのサブルーチンに飛ばします．
1030～1070	RANDOM	乱数で位置と色を決めて人を表示します．
1080～1240	CAR	車を表示してGETします．
1250～1620	RUNNING	人を3種類の形で動かして火を付けます．
1630～1970	MAN DROP	人の落ちていく動きを表示します．
1980～2640	LIFT	ハシゴの上下から端まできたときの判断，POINTの増減などを行ないます．
2650～2800	PUT OUT A FIRE	ヘリコプターと消火液の表示そして再開後の火の位置を決めて火を表示します．
2810～2880	GAME OVER	ビルを包む炎とGAME OVERの表示，そして少ししてからデモに戻ります．
2890～3050	DATA	3つに分割した炎とヘリコプターの消火液のデータが入っています．

デモ画面のフローチャート



ゲームはスピードが命という人には，ハシゴは文番号2030と2110と2160と2240を取り去り，車は1色（アトリビュートを使わない）で形を小さく変えれば少しは速くなるのではないかと思います．

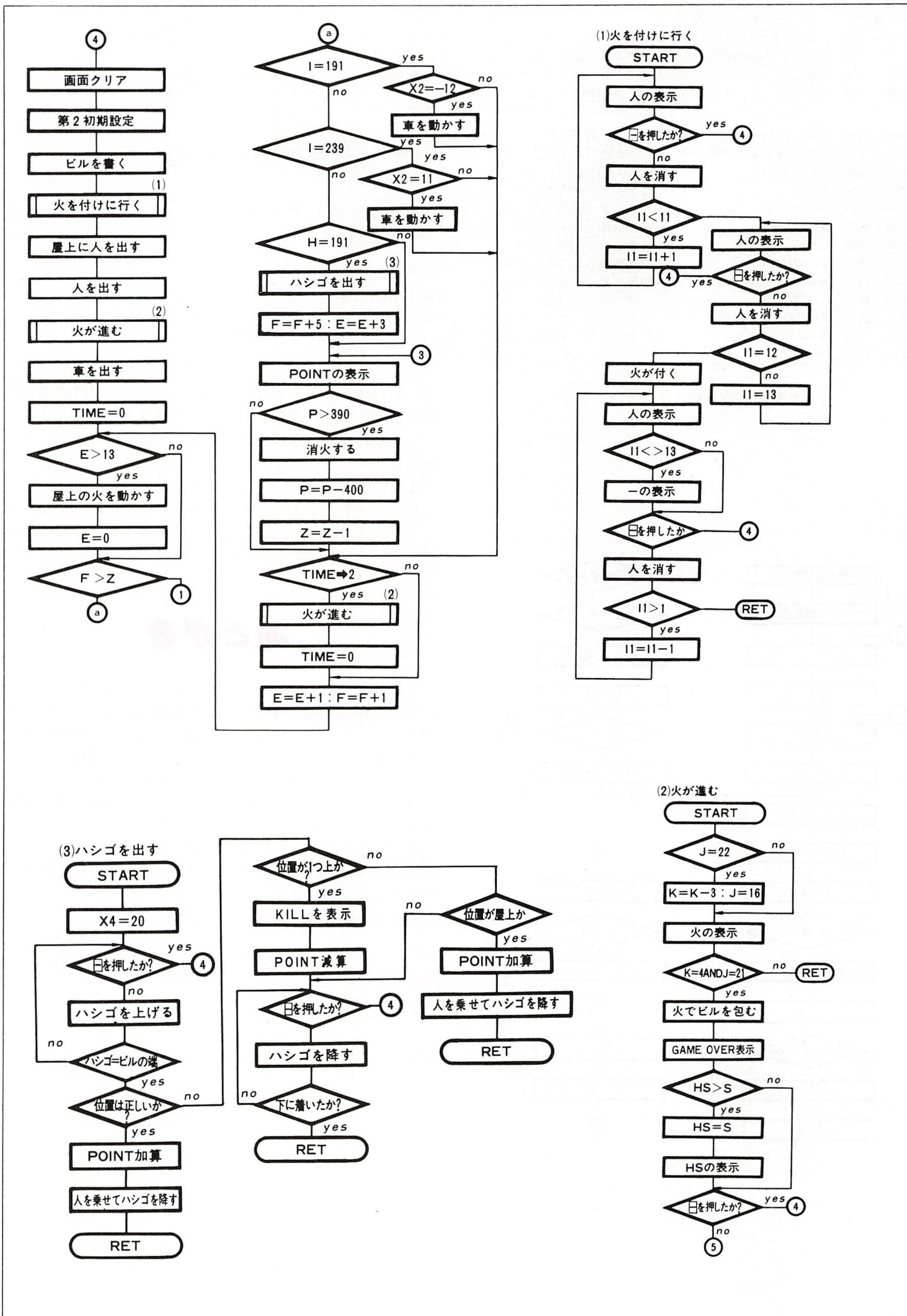
また，このゲーム方法にあきてきたら，人を助けるごとにG＝G＋1のような形で何人助けられたかという表示をしたり，人の落ちかたを速くしてデモのように落ちると火が進むという形に改造するのもいいのではないでしょうか．

あとがき

どうせゲームを作るならデモも本格的(？)に作ろうと思いデモを作ったら，量も多くなり時間もけっこうかかりました．このプログラムを作っていたころ近所で偶然火事があり，朝の4時ごろ，消防自動車のサイレンで起こされてしまいました．そのときのサイレンが迫力があったので，今度テープにとってB.G.M.としてゲームの最中に流すと緊迫感がしてくるのでは，などと考えています．

高得点をあげるためには，最初のうちは落ち着いて上の紫色の人はあまり狙わず，ゲームが進んでからカンを働かせてうまく救助するようにするといいでしょう．

ゲームのフローチャート



ファイア・レスキュー　プログラム・リスト

```
10 REM **** FIRE RESCUE ****
20 REM *** by S.NAKADAI ***
30 REM *** MAIN LOUTIN ***
40 CONSOLE,,0,1:WIDTH40,25:HS=0
50 REM *** DATA READ ***
60 DIM F1%(44),F2%(46),F3%(15),F4%(43),A1%(32),A2%(24),A3%(12),A4%(6),A5%(32),A6
%(24),A7%(12),A8%(6)
70 FOR T=0 TO 44:READ F1%(T):NEXT
80 FOR T=0 TO 46:READ F2%(T):NEXT
90 FOR T=0 TO 15:READ F3%(T):NEXT
100 FOR T=0 TO 43:READ F4%(T):NEXT
110 PRINT CHR$(12):GOSUB1090:GOTO 130
120 REM *** FIRST SCENE ***
130 PRINTCHR$(12):GOSUB 900:COLOR7
140 LOCATE 2, 6:PRINT "ｼﾞﾄﾞｳｼｬﾉ"
150 LOCATE 3, 7:PRINT "ｲﾄﾞｳ ｷｰﾊ"
160 LOCATE 4, 9:PRINT "4<->6"
170 LOCATE 2,11:PRINT "ﾊｼｺﾞﾉ ｷｰﾊ"
180 LOCATE 4,13:PRINT "SHIFT"
190 LOCATE 2,15:PRINT "GAME START"
200 LOCATE 2,17:PRINT "PUSH - ｷｰ"
210 COLOR3:LOCATE 26,6:PRINT "●  POINT"
220 LOCATE26,7:PRINT "+ SAFE 100"
230 COLOR4:LOCATE 26,9:PRINT "●  POINT"
240 LOCATE 26,10:PRINT "+ SAFE  40"
250 LOCATE 28,11:PRINT "OUT  -10"
260 LOCATE 28,12:PRINT "KILL -20"
270 COLOR6:LOCATE 26,14:PRINT "●  POINT"
280 LOCATE 26,15:PRINT "+ SAFE  20"
290 LOCATE 28,16:PRINT "OUT    0"
300 LOCATE 28,17:PRINT "KILL -10"
310 COLOR 7:LOCATE3,3:PRINT "Hi-SCORE"
320 LOCATE31,3:PRINT "SCORE"
330 LOCATE5,4:PRINT USING"#####";HS:LOCATE35,4:PRINT "0"
340 REM *** DEMO ***
350 GOSUB1260:X2=1:D=1
360 GOSUB690:J=17:K=22:S=0:P=0
370 GOSUB990:GOSUB1040:GOTO 380
380 GOSUB650:BEEP1:COLOR7:LOCATE31,4:BEEP0:PRINT USING"#####";S
390 GOSUB730:IF X2=-12 OR X2=11 THEND=-D:J=J+1:GOSUB990
400 IF X2=-9 OR X2=8 THENGOSUB1990:J=J+1:GOSUB990
410 IF R0=X2+11 THEN GOSUB1640:X2=X2+D:J=J+1:IF C0=0 THEN S=S-10:GOTO 370 ELSE 3
70
420 IF X2=0 THEN GOSUB1510:J=J+1:GOSUB990
430 FOR I=0TO30:IF INKEY$="-" THEN460 ELSE NEXT
440 X2=X2+D:GOTO 380
450 REM *** GAME ***
460 PRINTCHR$(12):S=0:P=0:E=0:F=0:Z=34:V1=0
470 COLOR7:LOCATE3,3:PRINT "Hi-SCORE"
480 LOCATE31,3:PRINT "SCORE"
490 LOCATE31,4:PRINTUSING"#####";S
500 LOCATE5,4:PRINTUSING"#####";HS
510 GOSUB900:GOSUB1260:GOSUB690:J=17:K=22
520 GOSUB990:GOSUB1040:X2=0
530 GOSUB730:TIME$="00:00:00"
540 IFE>13THENGOSUB1510:E=0
550 IFF>2THENBEEP1:GOSUB1640:BEEP0:C3=C0:GOSUB1040:F=0:IFC3=0THENS=S-10:P=P-10:G
OTO590
560 I=INP(0):H=INP(8):IFI=239THENIFX2=-12THEN610ELSEGOSUB650:X2=X2-1:BEEP1:GOSUB
730:BEEP0:GOTO610
570 IFI=191THENIFX2=11THEN610ELSEGOSUB650:X2=X2+1:BEEP1:GOSUB730:BEEP0:GOTO610
580 IFH=191THENGOSUB1990:E=E+3:F=F+5
590 COLOR7:LOCATE31,4:PRINTUSING"#####";S
600 IFP>390THENGOSUB2660:P=P-400:Z=Z-1
610 IFTIME$=>"00:00:02"THENTIME$="00:00:00":J=J+1:GOSUB990
620 E=E+1:F=F+1:GOTO0540
630 REM *** SUB LOUTIN ***
640 REM *** CAR PUT OUT ***
650 LINE(0,20)-(14,23)," ",BF
660 LINE(23,20)-(37,23)," ",BF
670 RETURN
680 REM *** TOP MAN FIRST POSITION ***
690 X5=19:COLOR3:LOCATEX5,4:PRINT "●"
700 LOCATEX5-1,5:PRINT "┴┴"
710 RETURN
720 REM *** CAR PUT ***
730 IF X2<-3 THEN X3=-4 ELSE X3=X2
740 ON X3+5 GOTO 760,770,790,810,830,850,870
```






```
750 PUT@ A(X2,20)-(X2+3,23),A1%:RETURN
760 PUT@ A(X2+35,20)-(X2+38,23),A5%:RETURN
770 PUT@ A(32,20)-(35,23),A5%
780 PUT@ A( 0,21)-( 0,23),A4%:RETURN
790 PUT@ A(33,20)-(36,23),A5%
800 PUT@ A( 0,21)-( 1,23),A3%:RETURN
810 PUT@ A(34,20)-(37,23),A5%
820 PUT@ A( 0,20)-( 2,23),A2%:RETURN
830 PUT@ A(35,20)-(37,23),A6%
840 PUT@ A( 0,20)-( 3,23),A1%:RETURN
850 PUT@ A(36,21)-(37,23),A7%
860 PUT@ A( 1,20)-( 4,23),A1%:RETURN
870 PUT@ A(37,21)-(37,23),A8%
880 PUT@ A( 2,20)-( 5,23),A1%:RETURN
890 REM *** BILL ***
900 COLOR7:LINE(0,24)-(39,24),"¯"
910 LINE(16,6)-(21,23),"■",BF
920 FOR J=9 TO 21 STEP 3
930 LOCATE15,J:PRINT "¯"
940 LOCATE22,J:PRINT "¯":NEXT
950 FOR I=7 TO 22 STEP3
960 COLOR1:LINE(16,I)-(21,I),"■":NEXT
970 RETURN
980 REM *** FIRE UP ***
990 IF J=22 THEN K=K-3:J=16
1000 COLOR2:LOCATEJ,K:PRINT "■"
1010 IF K=7 AND J=21 THEN GOSUB 2820:GOTO 130
1020 RETURN
1030 REM *** RANDOM  ***
1040 X0=INT(RND(1)*2):Y0=INT(RND(1)*4):C0=INT(RND(1)*2):R0=INT(RND(1)*22)
1050 X1=X0*7+15:Y1=Y0*3+7:C1=C0*2+4
1060 COLOR C1:LOCATEX1,Y1:PRINT "●"
1070 LOCATEX1,Y1+1:PRINT "+":RETURN
1080 REM *** CAR READ ***
1090 COLOR7:LINE(0,21)-(1,20),"/"
1100 LINE(36,20)-(37,21),"\"
1110 LOCATE 0,23:PRINT "o  o"
1120 LOCATE34,23:PRINT "o  o"
1130 COLOR2:LOCATE1,21:PRINT "| ◣"
1140 LOCATE34,21:PRINT "◢ |"
1150 LINE(0,22)-(3,22),"■":LINE(34,22)-(37,22),"■"
1160 LINE(1,23)-(2,23),"¯":LINE(35,23)-(36,23),"¯"
1170 GET@ A( 0,20)-( 3,23),A1%
1180 GET@ A( 1,20)-( 3,23),A2%
1190 GET@ A( 2,21)-( 3,23),A3%
1200 GET@ A( 3,21)-( 3,23),A4%
1210 GET@ A(34,20)-(37,23),A5%
1220 GET@ A(34,20)-(36,23),A6%
1230 GET@ A(34,21)-(35,23),A7%
1240 GET@ A(34,21)-(34,23),A8%:RETURN
1250 REM *** RUNNING ***
1260 I1=0
1270 COLOR2:LOCATEI1,21:BEEP1:PRINT "*":BEEP0
1280 COLOR7:LOCATEI1+1,21:PRINT "o"
1290 LOCATEI1,22:PRINT " └┼─"
1300 LOCATEI1,23:PRINT " └┴┐"
1310 FOR I=0TO40:IF INKEY$="-" THEN 460 ELSE NEXT
1320 LINE(I1,21)-(I1+3,23)," ",BF
1330 IF I1<=11 THENI1=I1+1:GOTO 1270
1340 I1=12:COLOR2:LOCATE 15,22:PRINT "♠"
1350 COLOR7:LOCATEI1+1,21:PRINT "o"
1360 LOCATEI1,22:PRINT " ┌┼─"
1370 LOCATEI1,23:PRINT " ┌┴┐"
1380 FOR I=0TO200:IF INKEY$="-" THEN 460 ELSE NEXT
1390 LINE(I1,22)-(I1+2,23)," ",BF:LOCATEI1+1,21:PRINT " "
1400 IF I1=12 THENI1=13:GOTO 1350
1410 COLOR2:LOCATE 16,22:PRINT "■"
1420 COLOR7:LOCATEI1,21:BEEP1:PRINT "o":BEEP0
1430 LOCATEI1-1,22:PRINT " └┼┐-"
1440 LOCATEI1-1,23:PRINT " ┌┴─"
1450 IF I1<>13 THEN LOCATEI1+3,23:PRINT "-"
1460 FOR I=0TO5:IF INKEY$="-" THEN 460 ELSE NEXT
1470 LINE(I1-1,22)-(15,23)," ",BF:LOCATEI1,21:PRINT " "
1480 IF I1>1 THENI1=I1-1:GOTO 1420
1490 RETURN
1500 REM *** TOP POSITION ***
1510 COLOR3:R1=INT(RND(1)*2)
1520 IF X5=16 AND R1=0 GOTO 1600
```

```
1530 IF X5=21 AND R1=1 GOTO 1600
1540 IF X5=17 AND R1=0 THEN X5=X5-1:GOTO 1600
1550 IF X5=20 AND R1=1 THEN X5=X5+1:GOTO 1600
1560 IF R1=0 THEN X5=X5-1 ELSE X5=X5+1
1570 LOCATEX5-1,4:PRINT "   ":LOCATEX5-2,5:PRINT "     "
1580 LOCATEX5,4:PRINT "●":LOCATEX5-1,5:PRINT "┴┴"
1590 RETURN
1600 LOCATEX5-1,4:PRINT "   ":LOCATEX5-2,5:PRINT "     "
1610 LOCATEX5,4:PRINT "●":LOCATEX5,5:PRINT "+"
1620 RETURN
1630 REM *** MAN DROP ***
1640 LINE(X1,Y1)-(X1,Y1+1)," ":IA=0
1650 COLORC1:IF X1=22 GOTO 1820
1660 LOCATEX1-1,Y1+1:PRINT "●"
1670 LOCATEX1-1,Y1+2:PRINT "+"
1680 LOCATEX1-1,Y1+1:PRINT " "
1690 IF IA=2 GOTO 1710
1700 Y1=Y1+1:IA=IA+1:GOTO 1660
1710 LOCATEX1-2,Y1+1:PRINT "  "
1720 IF Y1=19 AND X2>9 THEN RETURN
1730 IF IA=4 GOTO 1770
1740 LOCATEX1-2,Y1+2:PRINT "●+"
1750 Y1=Y1+1:IA=IA+1:GOTO 1710
1760 LINE(X1-2,Y1)-(X1-2,Y1+1)," "
1770 IF Y1=19 AND X2>9 THEN RETURN
1780 IF Y1=22 THEN RETURN
1790 LOCATEX1-2,Y1+1:PRINT "+"
1800 LOCATEX1-2,Y1+2:PRINT "●"
1810 Y1=Y1+1:GOTO 1760
1820 LOCATEX1+1,Y1+1:PRINT "●"
1830 LOCATEX1+1,Y1+2:PRINT "+"
1840 LOCATEX1+1,Y1+1:PRINT " "
1850 IF IA=2 GOTO 1870
1860 Y1=Y1+1:IA=IA+1:GOTO 1820
1870 LOCATEX1+1,Y1+1:PRINT "  "
1880 IF Y1=19 AND X2<-10 THEN RETURN
1890 IF IA=4 GOTO 1930
1900 LOCATEX1+1,Y1+2:PRINT "●+"
1910 Y1=Y1+1:IA=IA+1:GOTO 1870
1920 LINE(X1+2,Y1)-(X1+2,Y1+1)," "
1930 IF Y1=19 AND X2<-10 THEN RETURN
1940 IF Y1=22 THEN RETURN
1950 LOCATEX1+2,Y1+1:PRINT "+"
1960 LOCATEX1+2,Y1+2:PRINT "●"
1970 Y1=Y1+1:GOTO 1920
1980 REM *** LIFT ***
1990 Y4=20:IF X2<0 GOTO 2130
2000 X4=X2+1:COLOR7
2010 LOCATEX4,Y4:PRINT "/":LOCATEX4+1,Y4-1:PRINT "■"
2020 IF X4=13 GOTO 2050
2030 FOR I=0TO1:IF INKEY$="-" THEN 460 ELSE NEXT
2040 X4=X4+1:Y4=Y4-1:GOTO 2010
2050 IF X1=X4+2 AND Y1=Y4-3 GOTO 2260
2060 IF X1=X4+2 AND Y1=Y4-2 GOTO 2410
2070 IF X5=X4+3 AND Y4=8 GOTO 2510
2080 LOCATEX4,Y4:PRINT "/":LOCATEX4+1,Y4-1:PRINT "■"
2090 LOCATEX4+1,Y4-1:PRINT " "
2100 IF Y4=20 THEN RETURN
2110 FOR I=0TO1:IF INKEY$="-" THEN 460 ELSE NEXT
2120 X4=X4-1:Y4=Y4+1:GOTO 2080
2130 X4=X2+37:COLOR7
2140 LOCATEX4,Y4:PRINT "\":LOCATEX4-1,Y4-1:PRINT "■"
2150 IF X4=24 GOTO 2180
2160 FOR I=0TO1:IF INKEY$="-" THEN 460 ELSE NEXT
2170 X4=X4-1:Y4=Y4-1:GOTO 2140
2180 IF X1=X4-2 AND Y1=Y4-3 GOTO 2260
2190 IF X1=X4-2 AND Y1=Y4-2 GOTO 2460
2200 IF X5=X4-3 AND Y4=8 GOTO 2580
2210 LOCATEX4,Y4:PRINT "\":LOCATEX4-1,Y4-1:PRINT "■"
2220 LOCATEX4-1,Y4-1:PRINT " "
2230 IF Y4=20 THEN RETURN
2240 FOR I=0TO1:IF INKEY$="-" THEN 460 ELSE NEXT
2250 X4=X4+1:Y4=Y4+1:GOTO 2210
2260 BEEP1:LINE(X1,Y1)-(X1,Y1+1)," ":C2=C1:F=0
2270 IF C0=0 THEN S=S+40:P=P+40 ELSE S=S+20:P=P+20
2280 BEEP0:GOSUB1040:IF X4>20 GOTO 2350
2290 COLORC2:LOCATEX4+1,Y4-3:PRINT "●"
2300 LOCATEX4+1,Y4-2:PRINT "+"
```



ファイア・レスキュー　プログラム・リスト

```
2310 COLOR7:LOCATEX4,Y4:PRINT "／":LOCATEX4+1,Y4-1:PRINT "■"
2320 LINE(X4+1,Y4-3)-(X4+1,Y4-1)," "
2330 IF Y4=20 THEN RETURN
2340 X4=X4-1:Y4=Y4+1:GOTO 2290
2350 COLORC2:LOCATEX4-1,Y4-3:PRINT "●"
2360 LOCATEX4-1,Y4-2:PRINT "+"
2370 COLOR7:LOCATEX4,Y4:PRINT "＼":LOCATEX4-1,Y4-1:PRINT "■"
2380 LINE(X4-1,Y4-3)-(X4-1,Y4-1)," "
2390 IF Y4=20 THEN RETURN
2400 X4=X4+1:Y4=Y4+1:GOTO 2350
2410 LOCATEX4+1,Y4-1:PRINT "／":LOCATEX4-2,Y4-2:PRINT "KILL■"
2420 BEEP1:FOR I=0TO800:NEXT:BEEP0:LOCATEX4-2,Y4-2:PRINT "     "
2430 LOCATEX1,Y1+1:PRINT " "
2440 IF C0=0 THEN S=S-20:P=P-20 ELSE S=S-10:P=P-10
2450 F=0:GOSUB1040:GOTO 2080
2460 LOCATEX4-1,Y4-1:PRINT "＼":LOCATEX4-2,Y4-2:PRINT "■KILL"
2470 BEEP1:FOR I=0TO800:NEXT:BEEP0:LOCATEX4-2,Y4-2:PRINT "     "
2480 LOCATEX1,Y1+1:PRINT " "
2490 IF C0=0 THEN S=S-20:P=P-20 ELSE S=S-10:P=P-10
2500 F=0:GOSUB1040:GOTO 2210
2510 BEEP1:LINE(16,4)-(16,5)," ":COLOR3
2520 LOCATEX4+2,Y4-4:PRINT "●"
2530 LOCATEX4+2,Y4-3:PRINT "+"
2540 COLOR7:LOCATEX4+2,Y4-2:PRINT "■"
2550 LOCATEX4+1,Y4-1:PRINT "／"
2560 S=S+100:P=P+100:C2=3:BEEP0
2570 GOSUB690:LINE(15,4)-(15,6)," ":GOTO 2290
2580 BEEP1:LINE(21,4)-(21,5)," ":COLOR3
2590 LOCATEX4-2,Y4-4:PRINT "◕"
2600 LOCATEX4-2,Y4-3:PRINT "+"
2610 COLOR7:LOCATEX4-2,Y4-2:PRINT "■"
2620 LOCATEX4-1,Y4-1:PRINT "＼"
2630 S=S+100:P=P+100:C2=3:BEEP0
2640 GOSUB690:LINE(22,4)-(22,6)," ":GOTO 2350
2650 REM *** PUT OUT A FIRE ***
2660 COLOR7:LOCATEV1+1,0:PRINT "─┬─"
2670 LOCATEV1,1:PRINT "*▃○"
2680 LOCATEV1+2,2:PRINT "┴"
2690 COLOR5:IF V1>14 AND 25>V1 THEN PUT@(2*V1-6,8)-(2*V1+3,75),F4%,PSET
2700 FOR I=0TO200:NEXT
2710 IF V1>14 AND 25>V1 THEN PUT@(2*V1-6,8)-(2*V1+3,75),F4%,PRESET:FOR I=0TO200:
NEXT:GOSUB900
2720 BEEP1:LINE(V1,0)-(V1+3,2)," ",BF:BEEP0
2730 IF V1=34 THEN V1=0:GOTO 2750
2740 V1=V1+1:GOTO 2660
2750 GOSUB1040:COLOR2
2760 IF K=16 THEN LINE(16,22)-(19,22),"■":J=19:K=22:RETURN
2770 IF K=13 THEN LINE(16,22)-(21,22),"■":LOCATE16,19:PRINT "■":J=16:K=19:RETURN

2780 IF K=10 THEN LINE(16,22)-(21,22),"■":LINE(16,19)-(19,19),"■":J=19:K=19:RETU
RN
2790 IF K=7 THEN LINE(16,22)-(21,22),"■":LINE(16,19)-(21,19),"■":LOCATE16,16:PRI
NT "■":J=16:K=16:RETURN
2800 LOCATE16,22:PRINT "■":J=16:K=22:RETURN
2810 REM *** GAME OVER ***
2820 BEEP1:COLOR2,,1:PUT@(23,8)-(30,95),F1%,PSET
2830 PUT@(43,4)-(50,95),F2%;PSET
2840 PUT@(31,4)-(42,23),F3%,PSET
2850 COLOR7:LOCATE11,0:PRINT "GAME OVER":BEEP0
2860 IF HS<S THENHS=S:LOCATE5,4:PRINT  USING"#####";HS
2870 FOR I=0 TO 2000:IF INKEY$="-" GOTO 460 ELSE NEXT
2880 RETURN
2890 DATA 704,4128,14384,14392,31800,-900,-772,-1796,-1800,-1800
2900 DATA -7968,-3872,-2318,-2314,-772,-772,-772,-1796,-1800,-1800
2910 DATA -1800,-1800,-1800,-3848,-3856,-3856,-3856,-3342,-2314,-266
2920 DATA -772,-772,-772,-772,-772,-1796,-1286,-258,-258,-772
2930 DATA -772,-1800,-1800,-1800,-3856
2940 DATA 736,1028,1028,1558,9734,3847,3855,7967,7967,7967
2950 DATA 7967,3855,20303,20303,32623,32639,32639,16191,16191,16191
2960 DATA 16191,7967,7967,7967,7967,7967,20303,28495,32639,32639
2970 DATA 32639,16191,7999,7967,7967,3871,20303,20303,32639,32639
2980 DATA32639,32639,16191,16191,7999,7967,7967
2990 DATA 240,0,0,-32728,2050,2176,-16248,-21494,15041,-7188
3000 DATA -4226,32759,-1,-1,-1,-1
3010 DATA 760,512,8200,128,3075,-16336,768,10252,-32608,2562
3020 DATA -12236,832,12813,-24408,2690,-22486,17040,9482,18580,17698
3030 DATA 20756,-28372,-28124,4680,5193,17489,16656,2180,-31680,16912
3040 DATA 8456,2178,-30686,-32480,516,4100,64,1025,8208,128
3050 DATA 2050,16400,256,4
```





PC-8001 BASIC＋マシン語

抜群のスピードと迫力

STAR FIRE

雄城嘉史

東大ＰＣ-8001ユーザーズ・グループ

私は10ヵ月ほど前ＰＣを買い，マイコンを始めた男です．いままで『ＴＶゲームなぞなんだ！』と言い，意地でもＢＡＳＩＣにしがみついてきましたが，先日，突如Z80マシン語を学び始め，マシン語ゲームを作り始めました．

１作目は機械語のデフレクション・ゲームで、２作目がこの『ＳＴＡＲ　ＦＩＲＥ！』です．ハンド・アセンブルで作るガッツはないので，Z80全命令のアセンブル可能な１パス・アセンブラを自作して作り始めました（私はＤｉｓｋを持っていないのです）．

『どうせゲームを作るんだったら良いものを』と，市販ＴＶゲームの『スペース・シップ』をベースにしました．これはテーブル・タイプの名で，有名なゲームだと思いますが，ボックス・タイプの名が『スターファイア』です．

むろん同じゲームをＰＣ上にコピーしてもつまらないので，自分なりに考えてあります．思いの外簡単で，ゲームを考え始めてからおよそ半月で形はできてしまいました．

カラーグラフィックスでＰＣを上回る他機種では良く似たゲームがすでにあるそうですが，自分ではまぁまぁの線ではあろうというものができました．私も，私の妹もかなり熱くなってゲームに没頭し，目の前を宇宙船がいまも飛びかっております．

ゲームの説明

ゲームはわずかなＢＡＳＩＣ部（約１Ｋバイト）と，＆ＨＤ０００から使用する４Ｋバイト弱のマシン語部からなり，16Ｋシステムでももちろん走ります．

ＢＡＳＩＣはゲーム前後のタイトル表示などをわざわざマシン語で書くのがアホ臭かったのと，カラーモードの設定を行なわせるために書いたものです．ゲームの主要部はすべてマシン語で書かれています．

さて，ＲＵＮすると**写真１**のような画像が示されます．

ＳＴＡＲ　ＦＩＲＥ　！！！Ｖｅｒ．１

写真１　ゲーム・タイトルの表示

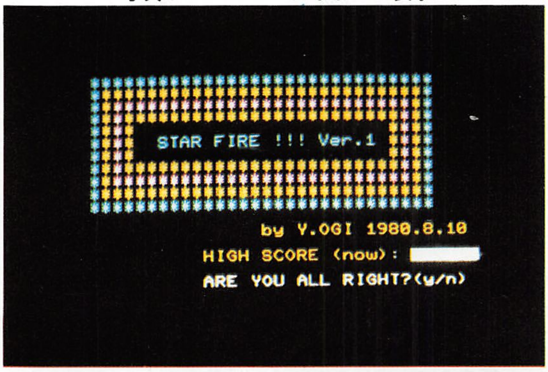
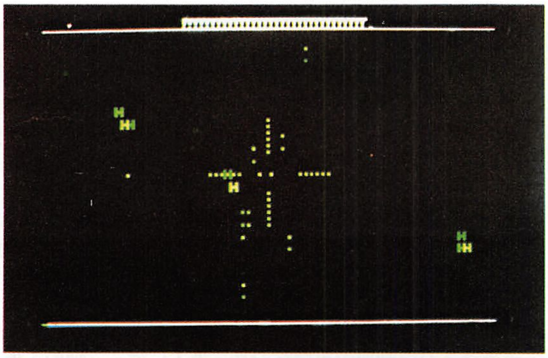


写真２　中央の十字が照準線



という文字の周囲を囲む星は様々な色に点滅し，高解像度カラーディスプレイなどで見るとなかなか美しいものです（このテクニックは東大五月祭で使いました）．また，画面にはそれまでに出された最高得点が示されます．

ゲームを開始するときは[Y]キーを押します．何らかの理由でゲーム開始をしないときは，他の任意のキーを押してください．両者ともに [RETURN] キーは不要です．

さて，[Y]キーが押されるとゲーム・スタートです．突如，**写真２**のような画面に切り換わります．

このゲームは要するに，プレイヤーが宇宙戦闘艇の唯一

図1　敵宇宙船およびミサイル

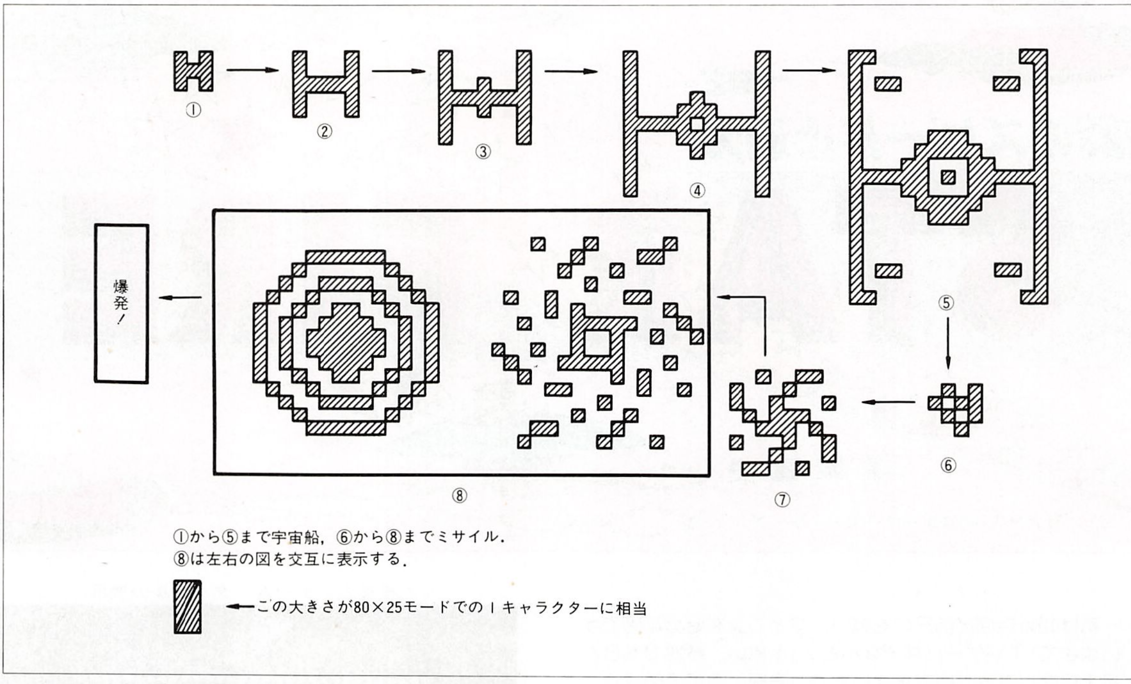


表1　画面の色と敵船1機あたりの得点

画面の色	1機当りの得点
緑	10点
水色	20点
黄	30点

の乗員となり，船とビーム砲を操って敵宇宙船を制限時間(変化します)内に破壊し，その得点を競うゲームです．

写真2で中央に十字に並んだ点が見えると思いますが，これが照準線です．上下の線の間がスコープで，中央上方，線のすぐ上に横たわる帯がタイマです．画面には星と敵宇宙船が動きまわります．

注意していただきたいのは，このゲームは画面の色を識別することを要求するということです．絶対にカラーでプレイしてください．画面の色が異なると敵宇宙船1機当たりの得点が異なります（**表1**）．

さて，少々解りにくいのは敵宇宙船がある程度接近してくると，自身をミサイル化して攻撃してくるという点です．敵宇宙船は5段階に大きくなりますが，一番大きな宇宙船はプレイヤー機が前進していると次に小さなミサイルに変じ，渦を巻きながら3段階に大きくなって(**図1**)，そして爆発します（**写真3**）．

爆発すると画面の色が変わり，**表1**に従って敵船1機当たりの得点が変化します．同時にタイマが進み残り時間が減少します．したがって，プレイヤーはもうこれ以上ミサイルをくらいたくないと考えるならば，バックするか，旋回して逃げるか，または撃ち落としてしまわねばなりません．

図2がキーボードからの操作法です．このゲームは複数のキー入力を受けつけます．したがって，テンキーの2つのキーを押せば機は斜めに旋回します．また，旋回しながらビーム砲を発射できます．

さて，プレイヤーはテンキーで機を旋回させ，敵機を画

図2　キーボードの説明

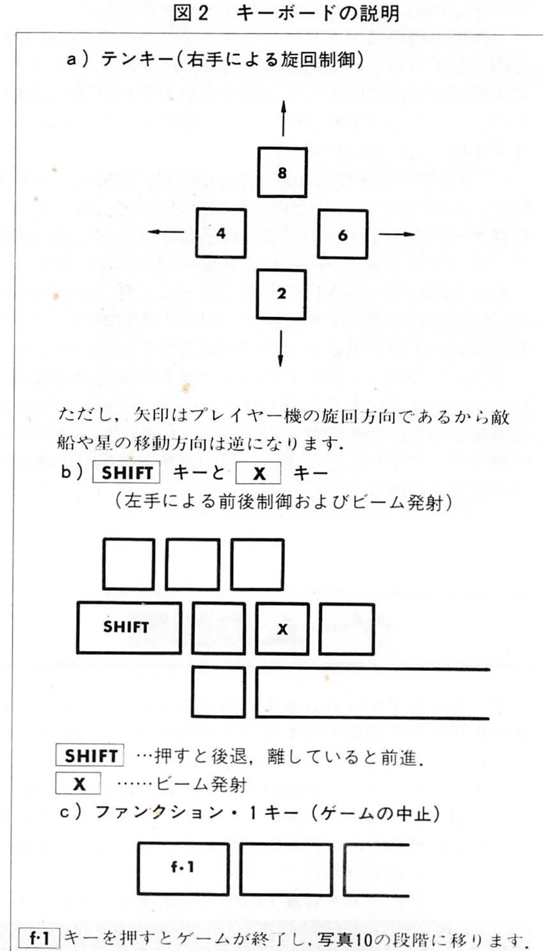


写真3　敵のミサイルが渦を巻きながら近ずいてくる

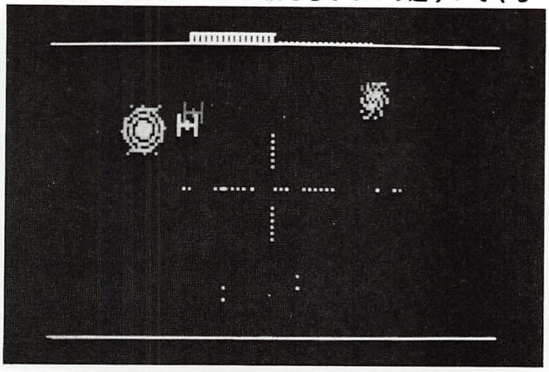


写真4　自動照準装置がロックしたところ

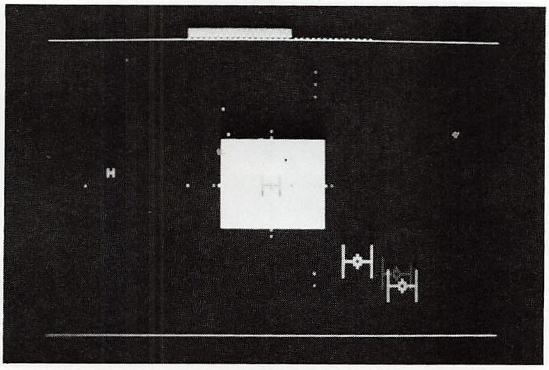


写真5　ビーム発射

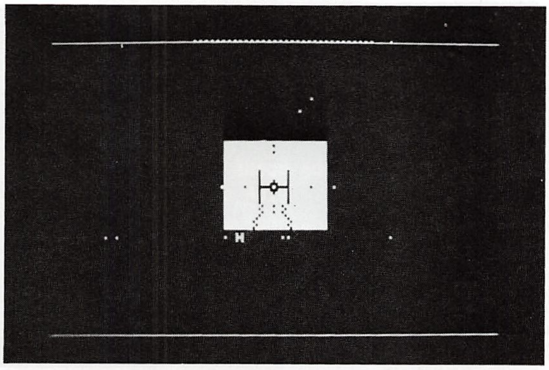


写真6　敵機がいないと，ビームが除々に消えていく

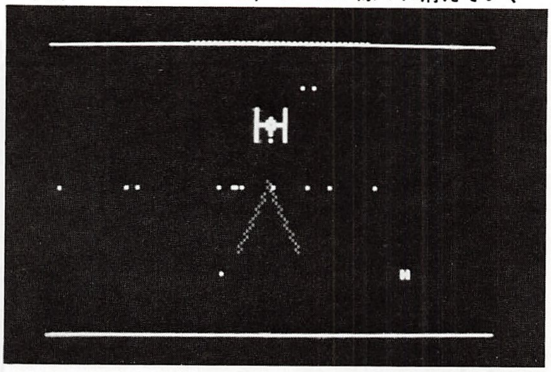


写真7　爆発パターン（ノーマル）

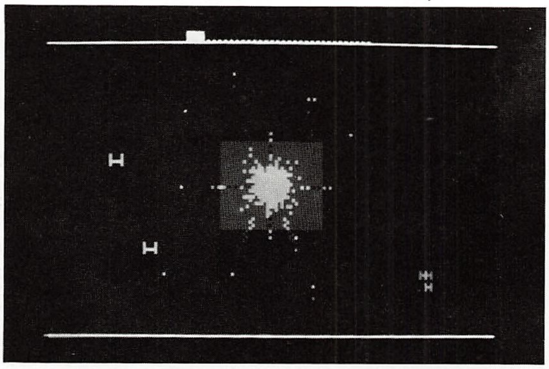
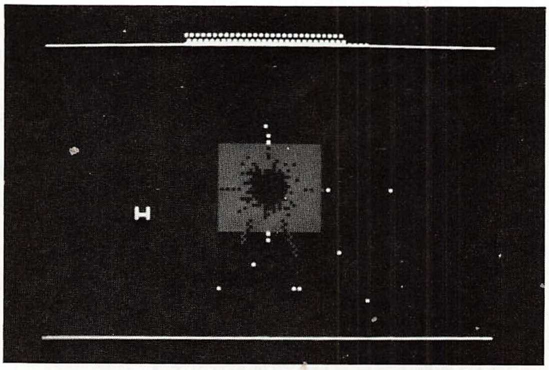


写真8　爆発パターン（リバース）



面中央，照準十字線の中央に持ってきます．すると自動照準装置が働いて，敵機はロックされます（**写真4**）．

自動照準装置は非常に強力な催眠暗示能力者で，『動くな！大きくも小さくもなるな！旋回したらついてこい！』と命令し，敵機は何があろうとロックから脱出できなくなります．プレイヤーはそこで**X**キーを押し，ビームを発射します．

ビームはＢＡＳＩＣの**ＬＩＮＥ**文と異なり，発射音をたてながら滑らかに伸びていきます．また，その間にも画面は動いています．ビームの色は画面が何色であろうと赤です．

ビームは中央に達すると，そこにロックされた機があれば破壊し，なければ何事も起こさず，再び滑らかに消えていきます．**写真5**にビーム発射直後の様子を，**写真6**に外れた場合にビームが消え始めた様子を示します．仮に命中した場合もビームの消え方そのものは変わりません．

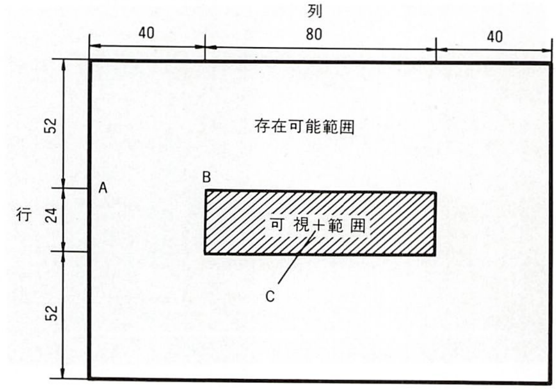
ビームが命中すると，爆発音がして中央は爆発の様子が赤と画面の色で,各々リバースとノーマルで点滅します(**写真7**，**写真8**)．爆発音はビームが消えるまで続き，その間**写真7**と**写真8**の状態が交互に映されるので，一応爆発らしく見えます．

このゲームでは１発のビームで複数機の敵機を破壊することが可能です．むろんロックするのは一度に１機ですがロックはその機にビームが届いた瞬間に解除され，次の敵機をロックできるようになります．

それに対し，ビームはある期間中央に達しています（すなわち，ビームの先端が中央に達してからビームが消えるまでの間．したがって，この期間中に次々に敵宇宙船を中央に持ってくればすべて爆破できるようになっています．

敵宇宙船がかたまって飛んでいたら，旋回しながらビー

図３　敵船の存在できる範囲とスコープ範囲



A：拡張座標系原点
B：画面内座標系原点
C：相対座標系原点
（画面中央点）

ムを放ち，なぎ倒すようにしてすべて１発で破壊することもできます．

ビーム砲で破壊したり，旋回したりして画面から敵宇宙船がいなくなると，画面の横に矢印が出現します（**写真９**）．敵宇宙船が存在できる範囲は**図３**に示すとおりスコープより広いため，どちら方に敵船がいるかをこの矢印が示してくれます．

ただし，矢印は左右だけで上下を示してはくれません．上下にも矢印を付けるのは簡単なのですが，プレイヤーの勘および記憶力が点数を左右する重要なファクターとなるべく，余り親切なことはしないようにしました．

敵ミサイルが視界外で爆発してもそれはプレイヤーに何の関係もなく，プレイヤーはそれを知ることもできません．

次にバックですが，SHIFT キーを押すと星々は中央に向け移動を始め，ミサイル化していない敵船は縮小し，やがて飛び去り，ミサイルは大きくなるのを止めます．SHIFT キーを押している間バックは続き，手を離すと前進モードに戻ります．

また，ミサイル攻撃を受けることのないバック・モードでの戦闘はプレイヤーに有利すぎ，またそんな戦法は卑怯だと考え，バック・モードでは一度有効範囲内に存在する敵船が０になると，バックし続ける限り新しく敵船をスコープ内に捕捉することは不可能になります．

つまり，**図３**の存在可能領域内に新しく敵船がワープ・アウトすることが決してなくなり，この範囲内に存在するミサイルや敵船を破壊し尽してしまったら，いくら待っても新しい獲物は現われません．

ゲームは上方のカウンタの帯グラフの長さが０になると終了で，**写真10**に示すとおり点数，敵船撃墜回数，被弾ミサイル数が表示され，特に得点がハイ・スコアであると**写真11**のようにその旨表示され，自動的に登録されます．

さて，ゲームの遊び方の最後にハイ・スコアのクリア方法を示します．普通プログラマブル・ファンクション・キーには15文字までしか登録できませんが，このゲームを走らせてから f･6 キーを押してください．20字以上の行が示され，ハイ・スコアをクリアします．

この方法はご存知の方も多いと思いますが，要するに f･6 キーを押しただけで，f･6 キーの内容と f･7 キー

写真９　画面内に敵機がいなくなると矢印が表われる

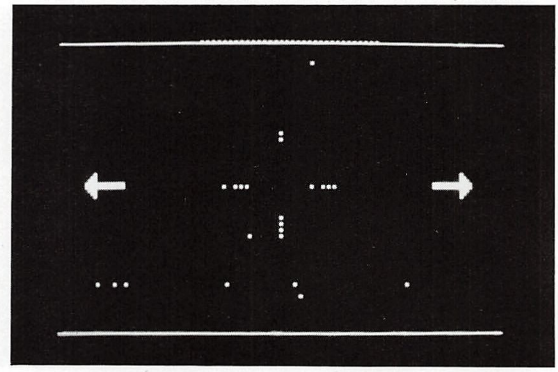


写真10　ゲーム・オーバーでスコアが表示される

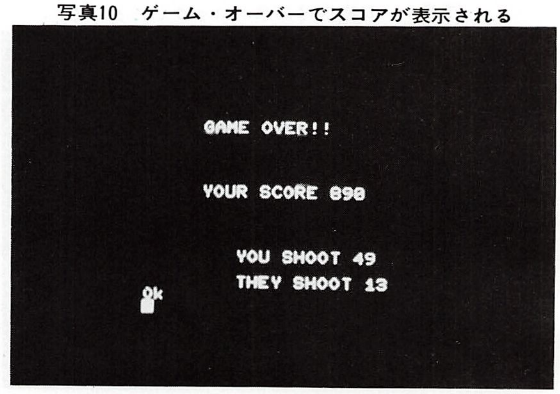
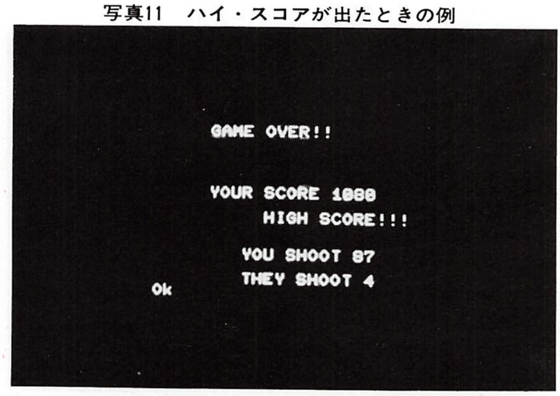


写真11　ハイ・スコアが出たときの例



の内容を連続して書かせているのです（ＫＥＹＬＩＳＴしてください）．

f･6 キーの内容はＲＡＭ上＆ＨＥＡＣＣ番地から16バイトに登録されていますが，通常の**ＫＥＹ**文では最後の＆ＨＥＡＤＢ番地は必ず＆Ｈ００が代入され，15文字以上は定義できないわけです．

この００Ｈによって f･6 と f･7 の境界は示されるのです．ファンクション・キーを押すと各キーの先頭番地から定義された内容は読まれるわけですが，読んでいくうちに００Ｈが出てくるとシステムはそこで読み込みを終了します．

では『００Ｈをなくしたらどうなるか』と考えて試行したところ，上のような使用法をみつけたわけです．ＢＡＳＩＣ部の150行を見てください．

プログラムの説明

ＢＡＳＩＣ部については特に説明の必要はないでしょう．マルチ・ステートメントの山で**ＲＥＭ**文もない，無茶苦茶に見づらいプログラムですが，そこを何とか読んでいただければわかると思います．要するに前後の文字表示のみ行なわせています．

次はマシン語ですが，　きわめて見づらいプログラムにこれもなっています．主要なワーキング・エリアは＆ＨＤ０００から＆ＨＤ21Ｆまでです（含データ領域)，キーインするときは＆ＨＤ０Ｃ８番地以後のみを入力してください．

敵宇宙船の総数は機の存在が許される範囲内で常に５機です．これに対し，星は画面内に10個存在します．**表２**に＆ＨＤ０００から＆ＨＤ０Ｃ７までの敵宇宙船および星に対応するワーキング・エリアの説明を，**表３**に主要な他のワーキング・エリアの説明を示します．

敵機や星の移動は各軸速度が**表２**に示す各軸方向の『時定数』に支配されており，船の接近や後退も同様です．**表４**はサブルーチンのうち主要なものの開始アドレスと簡単な説明です．**図４**に示すアルゴリズムと共に参照してください．

このゲームでは特にゲーム開始時にランダマイズを行なっていません，したがって，ロードして最初のゲームの最初のシーンでは同じ展開になる可能性がありますが，このゲームは人間のリアルタイムな反応が乱数に影響するようにしているので，事実上同じ展開が生じるとは思われません．いまのところハイスコアは9,400点です．

おわりに

ある日，私が行きつけの喫茶店に入ると店の中の配置が前日と少し違っていました．新しいテーブル・ゲームが入ったのです．それが『スペース・シップ』でした．

遊びもせずその前に腕を組んで座ること１時間．マシン語で次に何を作るか迷っていた私は１人でニヤけ始めました.『スターファイア』をPCで作ったろうじゃないか….

ゲームのデザインに10日，プログラム作成に３日，机上改良に３日，キーインに１日，デバッグに半日，グラフィック・デザイン改良に１日と半，他に少々いじって，計約20日弱で完成．後，同じクラスの，(あの)島田啓一郎君の助言により再びいじること半日で，このプログラムとなりました．

凄絶なほどの混乱型プログラムとなり，島田君の名誉のために申しておきますと，彼は主として『ゲームならそれらしく音とか爆発の様子とかつけろ！』とかいった面をアドバイスしてくれたのであって，混乱の責は私にあります．

ソース・プログラムを取っておけないノンディスク・ベースの１パス・アセンブラの悲しさでもありますが（改良＝ツギハギ！)，マシン語は確かに数値計算は苦手ですが，何としても皆さんアタックしてください．損になるわけはないし，思ったよりずっと簡単なものです．

なお，私のシステムは32Ｋ本体の他は黒白ＲＦモジュレータのみです．この私だってカラーのゲームが作れるのですから，他の人達はこれはもう何でもできるのでは….

ＰＣはI/Oポートやアトリビュート・エリアを始めとしてマシン語での直接制御はなかなか難しい機種ですが，相当にタフな機種でもあるので要はいじることです．

最後にかくもしょーもないプログラムを載せようとおっしゃっていただいたＩ/Ｏ誌と，Ｉ/Ｏ誌に載せるに際し，協力とゲーム・デザインに関し助言をいただいた東大ＴＳＧメンバーかつ東大ＰＣユーザーズ・グループメンバーの島田啓一郎君に対し，感謝の意を表したいと思います．

◘参考物件

1）テーブルタイプＴＶゲーム『スペース・シップ』

表２　敵宇宙船と星のためのワーキング・エリアの説明

a）敵宇宙船(５機分)

各宇宙船ごとのスタート・アドレス≡**IX**＝**& HD000**＋(宇宙船ナンバー－１)×**&H12**

ＩＸ＋００Ｈ	*拡張座標（x）(中心点)
０１Ｈ	〃　（y）(中心点)
０２Ｈ	*画面内座標（x）(中心点)
０３Ｈ	〃　（y）(中心点)
０４Ｈ	*相対座標絶対値（x）(中心点)
０５Ｈ	〃　（y）(中心点)
０６Ｈ	状態ナンバー(**図１**参照)
０７Ｈ	x方向移動時定数
０８Ｈ	y方向　〃
０９Ｈ	画像左上端画面内座標（x）
０ＡＨ	〃　（y）
０ＢＨ	状態変化時定数
０ＣＨ	状態ナンバー増分
０ＤＨ	状態変化時定数(オリジナル)
０ＥＨ	状態ナンバー増分(オリジナル)
０ＦＨ	状態変化用カウンタ
１０Ｈ	x方向移動用カウンタ
１１Ｈ	y方向　〃

b）星(10個分)

スタート・アドレス≡**IX**＝**& HD060**＋(星ナンバー－１)×**&H0A**

００Ｈ	*画面内座標（x）
０１Ｈ	〃　（y）
０２Ｈ	*相対座標絶対値（x）
０３Ｈ	〃　（y）
０４Ｈ	相対座標符号（x）
０５Ｈ	相対座標符号（y）
０６Ｈ	x方向移動時定数
０７Ｈ	y方向　〃
０８Ｈ	x方向カウンタ
０９Ｈ	y方向　〃

*）各座標系の原点は**図３**参照

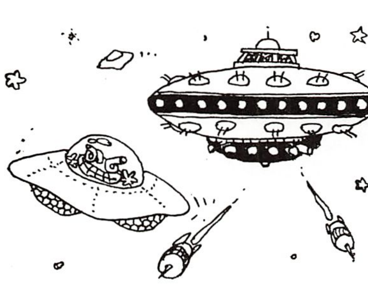


表３　主要なワーキング・エリア

番　地	内　　　　　容
Ｄ１８８Ｈ	敵船ナンバー（ＴＡＩＮＵＭ入力）
Ｄ１８ＤＨ Ｄ１８ＥＨ	敵船に対応するワーキング・エリアの開始番地（ＴＡＩＮＵＭ出力）
Ｄ１８ＦＨ	星ナンバー（ＳＴＲＮＵＭ入力）
Ｄ１９４Ｈ Ｄ１９５Ｈ	星に対するワーキング・エリアの開始番地（ＳＴＲＮＵＭ出力）
Ｄ１９６Ｈ	消したり描いたりする画像の左上端x座標
Ｄ１９７Ｈ	同y座標(PSET, PRESET入力)（上と共に）
Ｄ１９８Ｈ Ｄ１９９Ｈ	画像データ開始番地（同上）
Ｄ１Ａ０Ｈ	描こうとした図形が表示可能な範囲にあったかどうか.
Ｄ１Ｄ４Ｈ Ｄ１Ｄ５Ｈ	ＲＮＤ数作成用ＲＯＭ読みポインタ
Ｄ１Ｄ６Ｈ Ｄ１Ｄ７Ｈ	ゲームの総力カウンタ（&H２０００以上となるとゲーム終了）
Ｄ１Ｅ０Ｈ	画面の色を決定するためのアトリビュート・エリア代入値
Ｄ１Ｅ１Ｈ	ビーム砲制御．０でビームなし１でビーム表示，２で消去
Ｄ１Ｆ４Ｈ	ゲーム終了判定用
Ｄ１Ｆ６Ｈ Ｄ１Ｆ７Ｈ	画面座標x（ＰＯＩＮＴ入力） 画面座標y（ＰＯＩＮＴ入力）
Ｄ１Ｆ８Ｈ Ｄ１Ｆ９Ｈ	上の２入力に対応するV-RAM内番地（ＰＯＩＮＴ出力）
Ｄ１ＦＡＨ	ロックされた敵船ナンバー
Ｄ２０６Ｈ Ｄ２０７Ｈ	得点
Ｄ２０ＡＨ	画面のクリア・コード（ＣＬＥＡＲ入力）
Ｄ２０ＣＨ	敵ミサイルの爆発回数
Ｄ２０ＤＨ Ｄ２０ＥＨ	ハイ・スコア
Ｄ２１１Ｈ Ｄ２１２Ｈ	撃墜した敵船の数

表４　主要なサブルーチン

開始番地	名　前	説　　　明
Ｄ２２０Ｈ	POINT	正負表現１バイトのx，y（画面内）座標から対応するV-RAM内アドレスを算出する.
Ｄ２５４Ｈ	PUT	左上端の画面内座標と画像データの先頭番地を与えられ，画像を描くか，または消すかする.
Ｄ２Ｆ０Ｈ	PSET	ＰＵＴの属性を描くべく指定しコールする.
Ｄ３００Ｈ	PRESET	ＰＵＴの属性を消去すべく指定しコールする.
Ｄ３１０Ｈ	TAINUM	敵船ナンバーを与えられ該当するワーキング・エリア先頭番地，およびその船の状態から画像データ先頭番地を出力する.
Ｄ３４４Ｈ	SRND	ＲＯＭを読み込みＲＮＤ数を作成する.
Ｄ３８０Ｈ	RNTAI	敵船のワープ・アウト先座標を定める.
Ｄ３Ｆ０Ｈ	EDGE	敵船の中心拡張座標から他の座標を定める.
Ｄ４５０Ｈ	ATTRIB	Ｄ１Ｅ０Ｈに与えられた数値をアトリビュート・エリア各行左から２番目のアドレスに代入.
Ｄ４８０Ｈ	PREPA	アトリビュート・エリアにすべて０を代入する.
Ｄ４Ｃ０Ｈ	CLEAR	与えられたコードで画面を埋める.
Ｄ５００Ｈ	CENTER	画面中央の色やモードを指定されたように変える.
Ｄ５５０Ｈ	LOCK	色は白，モードはリバースとしてＣＥＮＴＥＲをコールする.
Ｄ５６０Ｈ	UNLOCK	画面中央の色とモードを元に戻せとＣＥＮＴＥＲに命令する.
Ｄ５７０Ｈ	REDZ	色は赤，モードはリバースとしてＣＥＮＴＥＲをコールする.
Ｄ５８０Ｈ	BOMB	ＣＬＥＡＲ，ＰＲＥＰＡ，ＡＴＴＲＩＢ，ＳＯＵＮＤを使い，敵ミサイルの爆発を表現し画面の色を変えカウンタを進める.
Ｄ５Ｆ０Ｈ	TAIMOV	敵船の移動，変化，ロック，出現など敵船に関するすべてを制御する（ミサイル燃発も）.
Ｄ７４０Ｈ	ARROW	左右の矢印を描いたり消したりする.
Ｄ７Ｄ８Ｈ	STRNUM	星ナンバーから該当するワーキング・エリアの先頭番地を算出する.
Ｄ７Ｆ８Ｈ	RNSTR	星の出現場所を定める.
Ｄ８８８Ｈ	STEDGE	星の画面内座標から他の座標を定める.
Ｄ８Ｆ０Ｈ	CONST	星の移動時定数を定める.
Ｄ９０８Ｈ	STRMOV	星に関するすべてを制御する.
Ｄ９Ｅ５Ｈ	BEAMCN	ビームに関するすべてを制御する（敵船爆発など．なお，敵ミサイルの爆発はビームを妨害する）.
ＤＢ３０Ｈ	CROSS	中央の十字線およびカウンタをつかさどる.
ＤＢ９０Ｈ	INPUT	キーボードからの入力をチェックし，必要な処理を行なう.
ＤＣ５０Ｈ	PREPTA	ゲーム開始前の初期化を行なう.
ＤＣＡ０Ｈ	MAIN	各サブルーチンを順序だててコールし，Ｄ１Ｆ４Ｈが０である限り繰り返す.
ＤＤ５０Ｈ	SOUND	敵ミサイル命中時の音出しルーチン.

図4　フローチャート(部分)

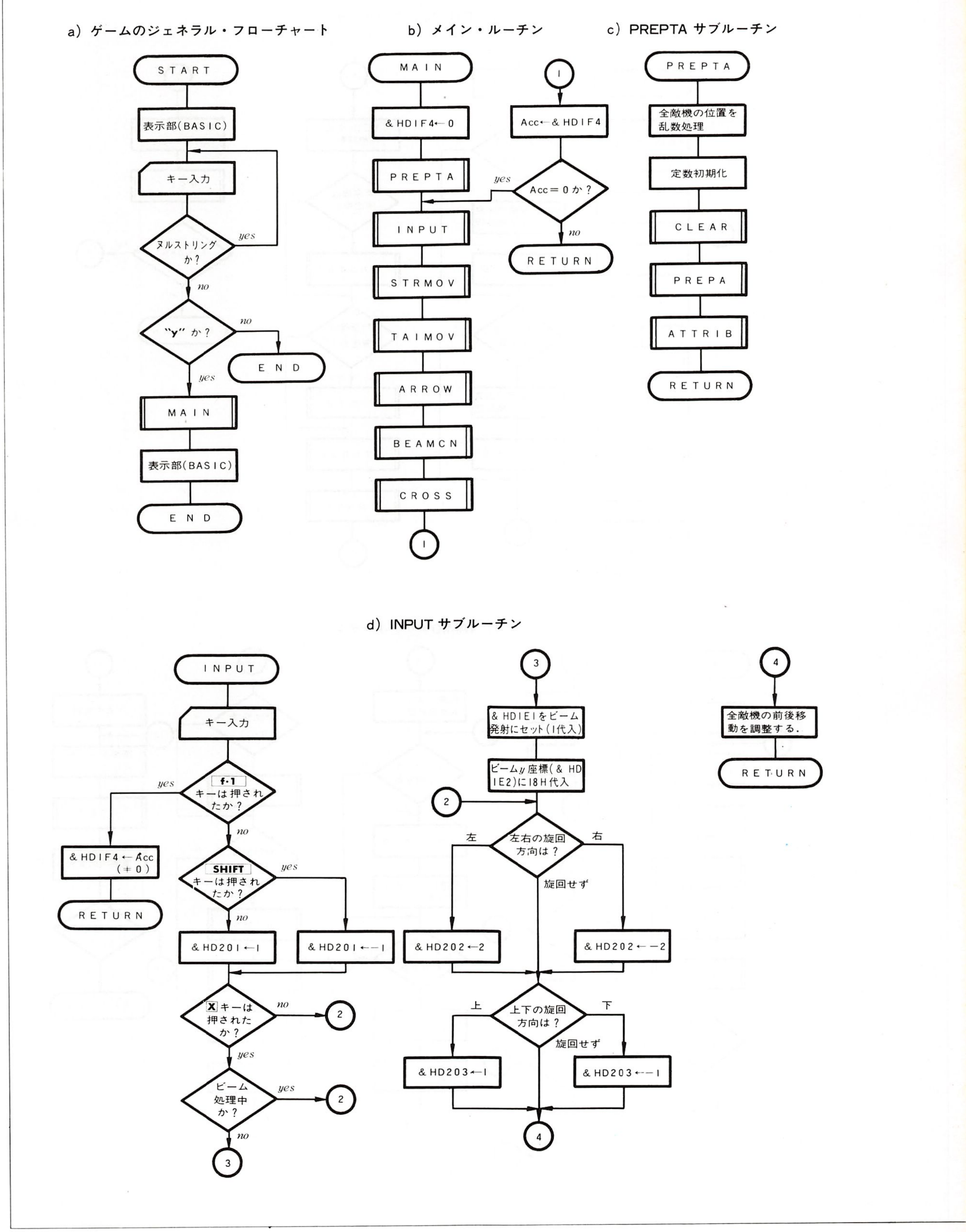
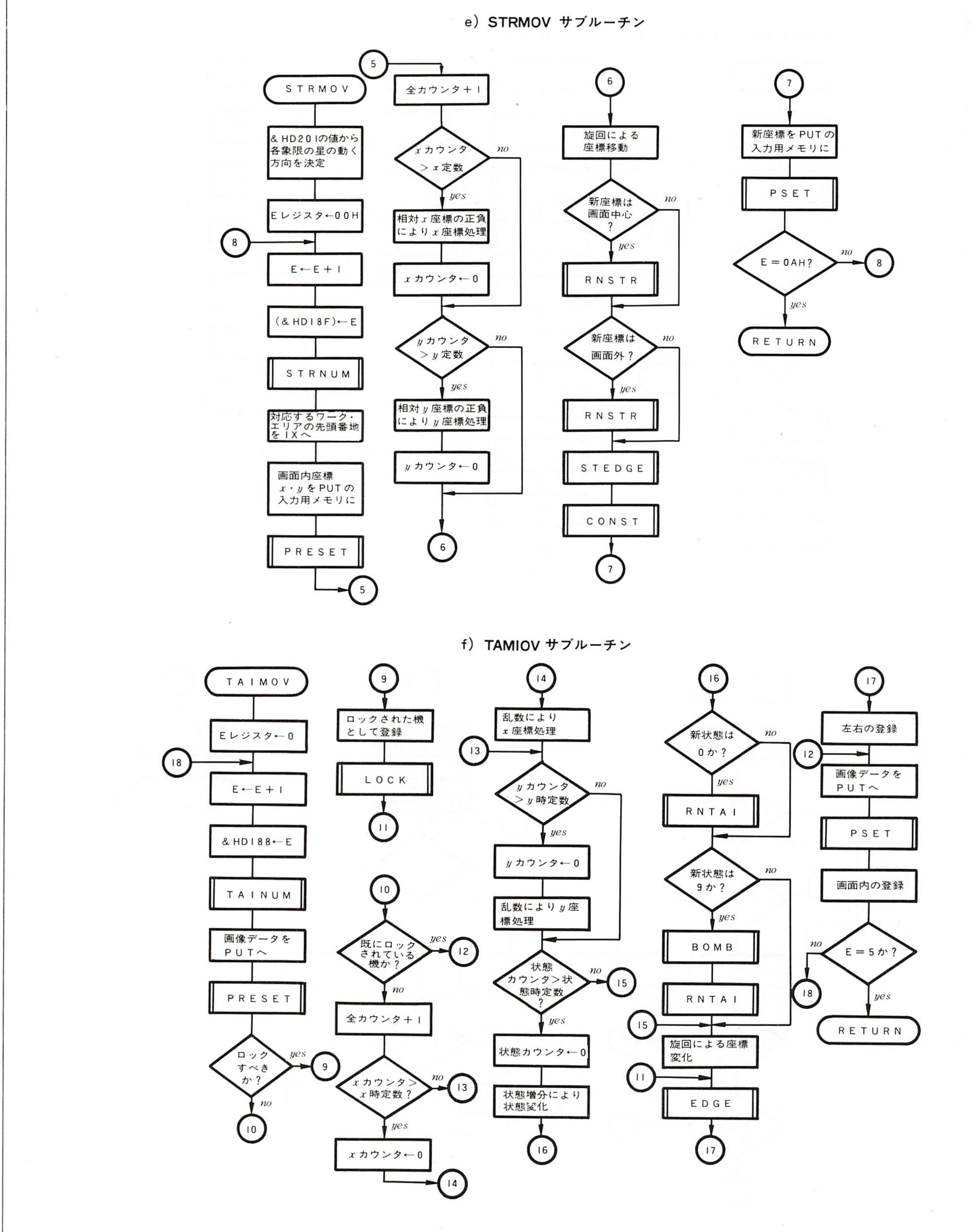
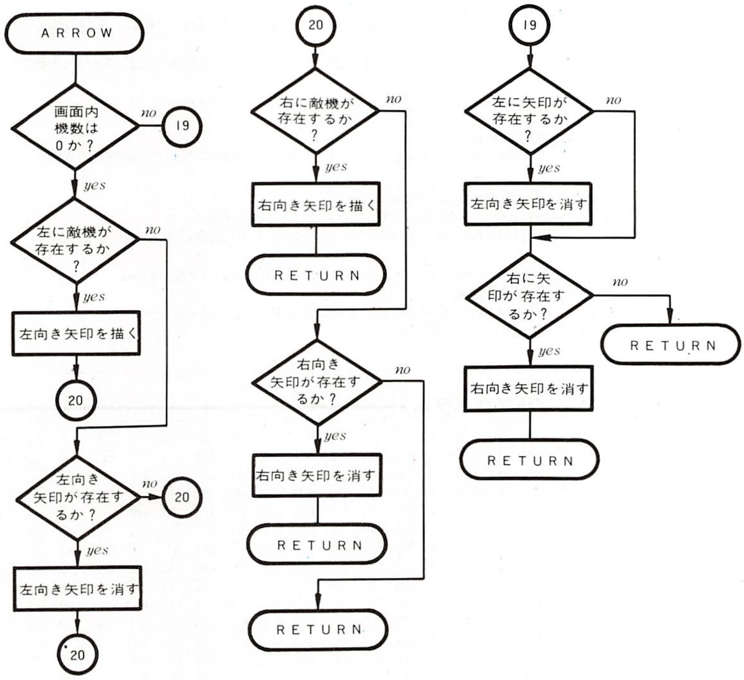


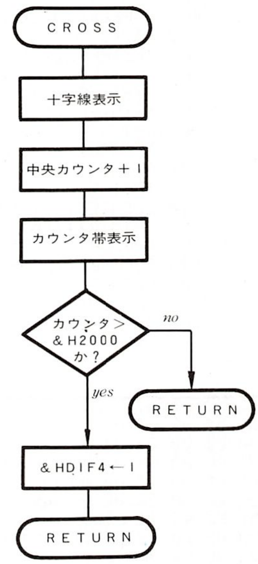
図4　フローチャート（部分）



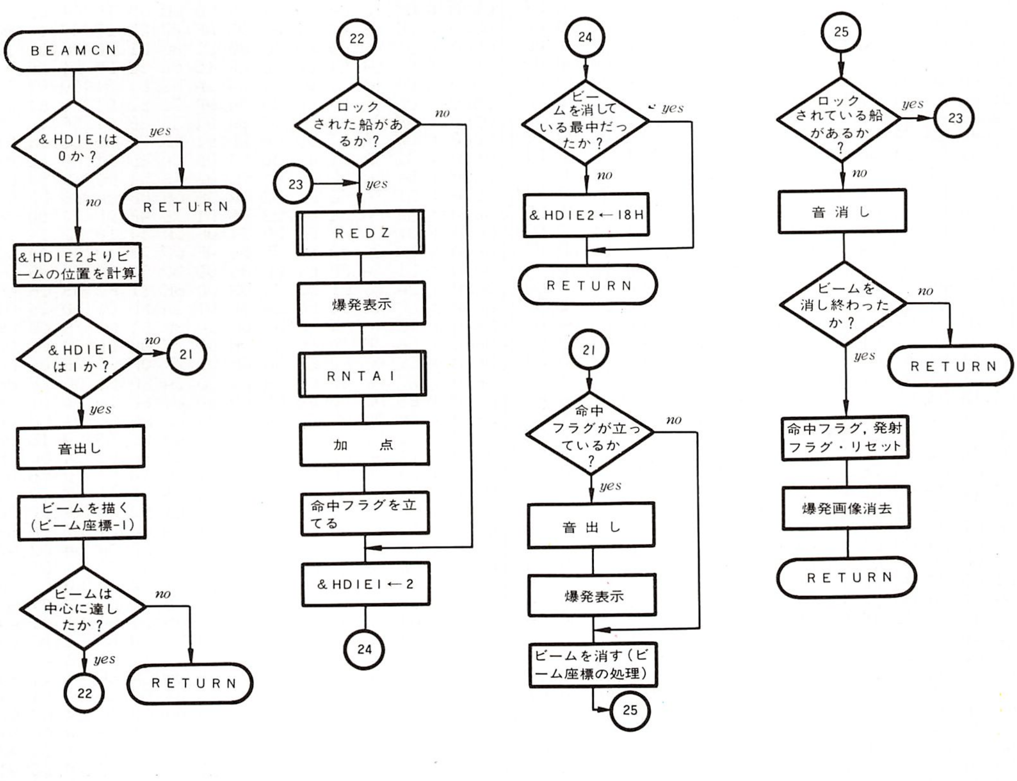
g）ARROW サブルーチン



i）CROSS サブルーチン



h）BEAMCN サブルーチン



BASIC プログラム

```
10 CLEAR300,&HCFFF:DEFUSR0=&HDCA0:WIDTH36,25:CONSOLE0,25,0,1:COLOR7,,1:PRINTCHR$
(12)
20 LINE(3,3)-(26,7),"*",2,B:LOCATE5,5:COLOR5:PRINT" STAR FIRE !!! Ver.1 ":LOCATE
15,12:COLOR6:PRINT"by Y.OGI 1980.8.10"
30 LOCATE10,14:PRINT"HIGH SCORE (now): ";
40 A0=PEEK(&HD20E)*256+PEEK(&HD20D):PRINTA0:LOCATE10,16:COLOR7:PRINT"ARE YOU ALL
 RIGHT?(y/n)"
50 POKE&HF9E2,1:POKE&HF9E3,&HC8:POKE&HF9E4,56:POKE&HF9E5,&HE8:POKE&HF9E6,59:POKE
&HF9E7,6:POKE&HF9E8,72:POKE&HF9E9,0
60 AX=FIX(RND(1)*4)+1:AC=FIX(7*RND(1))+1:LINE(4-AX,4-AX)-(25+AX,6+AX),"*",AC,B
70 S$=INKEY$:IFS$=""THEN60
80 IFS$<>"y"THENENDELSEWIDTH80,25:A=USR0(A)
90 WIDTH36,25:CONSOLE0,20,0,0:COLOR0,,0:PRINTCHR$(12):LOCATE5,5:PRINT"GAME OVER!
!";CHR$(7)
100 IF(INP(0)ANDINP(1)ANDINP(5)ANDINP(9))<>255THEN100
110 LOCATE5,10:PRINT"YOUR SCORE";:A1=PEEK(&HD207)*256+PEEK(&HD206)
120 PRINTA1:IFA1>A0THENLOCATE10,12:PRINT"HIGH SCORE!!!":POKE&HD20D,PEEK(&HD206):
POKE&HD20E,PEEK(&HD207)
130 LOCATE8,15:PRINT"YOU SHOOT";:A2=PEEK(&HD212)*256+PEEK(&HD211):PRINTA2
140 LOCATE8,17:PRINT"THEY SHOOT";PEEK(&HD20C)
150 KEY6,"poke&Hd20d,0:po":KEY7,"e&Hd20e,0"+CHR$(13):POKE&HEADB,&H6B:END
160 INPUTA0:POKE&HD20E,A0¥256:POKE&HD20D,A0MOD256
```

マシン語プログラム

```
D0C8 : 02 01 4E 0E 03 01 F0 44
D0D0 : F4 00 00 00 04 03 08 00 00 08 4F E4 44 0F 03 00
D0E0 : 00 03 00 00 06 03 0E 00 00 00 00 0E 4F 44 BE 4E
D0F0 : 44 0F 0F 00 10 00 09 0F 00 00 00 00 08 05 2E 88
D100 : 00 00 00 80 28 0E 0F 00 00 88 08 00 00 0F 4F 44
D110 : FE 51 F1 4E 44 0F 0F 00 10 33 13 00 00 0F 6F 22
D120 : 00 00 00 20 82 0F 00 00 00 00 00 00 02 01 52 7A
D130 : 04 02 86 C1 9A 41 82 59 83 61 00 00 00 00 00 00
D140 : 08 04 00 10 40 92 40 20 12 00 10 04 03 4E C4 15
D150 : 28 84 21 14 A8 23 72 C0 20 08 00 48 04 02 49 02
D160 : 08 00 00 00 00 00 00 00 00 01 01 02 03 00 01 03
D170 : 00 00 01 01 02 00 01 01 C8 D0 CC D0 D4 D0 E4 D0
D180 : FC D0 2C D1 30 D1 40 D1 05 02 02 02 0A 48 D0 0A
D190 : FF 01 FF 01 BA D0 E4 0B CC D0 03 01 00 01 E4 14
D1A0 : 00 01 01 04 07 03 00 C8 00 00 00 00 00 F6 FF 66
D1B0 : 66 66 66 66 00 31 00 00 00 00 00 00 07 03 00 00
D1C0 : 00 00 00 8C 00 66 66 66 66 66 FF 6F 00 00 00 00
D1D0 : 00 13 00 00 CF 1B 6E 05 01 00 0A 98 14 B8 1E D8
D1E0 : 98 00 0C 01 00 1E F8 1F 04 31 00 32 00 00 00 00
D1F0 : 00 00 00 00 FD 00 E4 13 CC FB 04 36 D0 FF 01 0D
D200 : 50 01 00 00 00 00 9A 01 D0 D0 00 00 00 00 00 00
D210 : 00 29 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D220 : 00 21 00 F3 01 78 00 3A F7 D1 A7 CA 3F D2 CB 7F
D230 : CA 3A D2 A7 ED 42 3C C3 2A D2 09 3D C3 2A D2 3A
D240 : F6 D1 4F CB 7F CA 4D D2 06 FF C3 4F D2 06 00 09
D250 : 22 F8 D1 C9 21 00 00 22 9C D1 2A 96 D1 22 9E D1
D260 : DD E5 DD 21 96 D1 AF DD 77 0A 2A 98 D1 7E DD 77
D270 : 04 23 7E DD 77 05 22 08 D2 2A 08 D2 DD 7E 09 FE
D280 : 17 F2 CA D2 FE 02 FA CA D2 DD 7E 00 32 F6 D1 DD
D290 : 7E 09 32 F7 D1 E5 CD 20 D2 E1 ED 4B F8 D1 23 DD
D2A0 : 7E 08 FE 4F F2 BA D2 FE 02 FA BA D2 3E 01 DD 77
D2B0 : 0A 3A E3 D1 A7 CA B9 D2 7E 02 03 DD 34 06 DD 34
D2C0 : 08 DD 7E 04 DD BE 06 C2 9E D2 DD 7E 00 DD 77 08
D2D0 : AF DD 77 06 DD 34 07 DD 34 09 CD E0 DC DD 7E 05
D2E0 : DD BE 07 C2 79 D2 DD E1 C9 00 00 00 00 00 00 00
D2F0 : 00 3E 01 32 E3 D1 CD 54 D2 C9 00 00 00 00 00 00
D300 : 00 AF 32 E3 D1 CD 54 D2 C9 00 00 00 00 00 00 00
D310 : 00 3A 88 D1 FD 21 00 D0 01 12 00 3D CA 24 D3 FD
D320 : 09 C3 1B D3 FD 22 8D D1 FD 4E 06 CB 21 06 00 FD
D330 : 21 76 D1 FD 09 FD 4E 00 FD 46 01 ED 43 98 D1 CD
D340 : E8 DD C9 00 00 2A D4 D1 ED 4B D6 D1 09 CB BC 7E
D350 : E6 02 3D 32 FD D1 23 7E E6 07 32 FE D1 23 7E E6
D360 : 0F 32 FF D1 23 7E E6 77 32 00 D2 0A 4F 06 00 09
D370 : 22 D4 D1 C9 00 00 00 00 00 00 00 00 00 00 00 00
D380 : 00 DD 2A 8D D1 CD 44 D3 3A FF D1 C6 05 DD 77 01
D390 : 3A FE D1 DD 77 07 3A 00 D2 57 CD 44 D3 3A FE D1
D3A0 : DD 77 08 3A FD D1 FE 01 C2 AF D3 3E 28 82 57 7A
D3B0 : DD 77 00 CD 44 D3 3A FF D1 DD 77 0B DD 77 0D 3A
D3C0 : 01 D2 DD 77 0C DD 77 0E 3E 01 DD 77 06 AF DD 77
D3D0 : 0F DD 77 10 DD 77 11 CD C1 DC CD F0 D3 C9 00 00
D3E0 : AF 32 12 D2 32 10 D2 CD 50 DC C9 00 00 00 00 00
D3F0 : 00 FD 2A 8D D1 FD 7E 01 FD 77 03 FD 7E 00 D6 28
D400 : FD 77 02 FE 27 F2 0F D4 47 3E 27 90 C3 11 D4 D6
D410 : 27 FD 77 04 FD 7E 03 FE 0C F2 23 D4 47 3E 0C 90
D420 : C3 25 D4 D6 0C FD 77 05 3E 67 FD 86 06 6F 26 D1
D430 : 46 FD 7E 02 90 FD 77 09 7D C6 08 6F 46 DD 7E 03
D440 : 90 DD 77 0A C9 00 00 00 00 00 00 00 00 00 00 00
D450 : 00 AF 32 F7 D1 3E 51 32 F6 D1 CD 20 D2 2A F8 D1
D460 : 3A E0 D1 77 3A F7 D1 3C 32 F7 D1 FE 19 C2 5A D4
D470 : C9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D480 : D5 21 4F F3 AF 4F 3C 23 71 FE 27 C2 86 D4 AF 32
D490 : F7 D1 3E 50 32 F6 D1 21 F7 D1 34 CD 20 D2 ED 5B
D4A0 : F8 D1 21 50 F3 01 27 00 ED B0 3A F7 D1 FE 18 C2
D4B0 : 97 D4 D1 C9 00 00 00 00 00 00 00 00 00 00 00 00
D4C0 : D5 21 FF F2 3A 0A D2 4F AF 3C 23 71 FE 4F C2 C9
D4D0 : D4 AF 32 F6 D1 32 F7 D1 21 F7 D1 34 CD 20 D2 ED
D4E0 : 5B F8 D1 21 00 F3 01 4F 00 ED B0 3A F7 D1 FE 18
D4F0 : C2 D8 D4 D1 C9 00 00 00 00 00 00 00 00 00 00 00
D500 : D5 3E 08 32 F7 D1 3E 52 32 F6 D1 21 F7 D1 34 CD
D510 : 20 D2 ED 5B F8 D1 01 07 00 3A D8 D1 A7 CA 3E D5
D520 : 21 E5 D1 ED B0 FD 2A F8 D1 3A E0 D1 FD 77 07 3A
D530 : E4 D1 A7 CA 44 D5 3E 58 FD 77 01 C3 44 D5 21 EC
D540 : D1 03 ED B0 3A F7 D1 FE 0F C2 0B D5 D1 C9 00 00
D550 : 00 3E 01 32 D8 D1 AF 32 E4 D1 CD 00 D5 C9 00 00
D560 : 00 AF 00 00 00 32 D8 D1 CD 00 D5 C9 00 00 00 00
D570 : 00 3E 01 32 E4 D1 32 D8 D1 CD 00 D5 C9 00 00 00
D580 : 3A A0 D1 A7 C8 CD 80 D4 21 39 DD 3E FF 32 0A D2
D590 : E5 CD C0 D4 E1 CD 50 DD D5 16 04 E5 21 D9 D1 34
D5A0 : 7E FE 03 C2 A8 D5 AF 77 CB 27 4F 06 00 FD 21 DA
D5B0 : D1 FD 09 FD 7E 00 32 8C D1 FD 7E 01 32 E0 D1 CD
D5C0 : 50 D4 E1 CD 50 DD 15 C2 9B D5 D1 AF 32 0A D2 CD
D5D0 : C0 D4 21 0C D2 34 2A D6 D1 01 00 01 09 22 D6 D1
D5E0 : CD 80 D4 CD 50 D4 AF 32 E2 D1 32 E1 D1 C9 00 00
D5F0 : AF 32 89 D1 32 8A D1 32 8B D1 5F 1C 7B 32 88 D1
D600 : CD 10 D3 DD 2A 8D D1 DD 7E 09 32 96 D1 DD 7E 0A
D610 : 32 97 D1 CD 00 D3 DD 2A 8D D1 DD 7E 04 FE 03 F2
D620 : 54 D6 DD 7E 05 FE 02 F2 54 D6 CD 50 D5 DD 2A 8D
D630 : D1 3A FA D1 A7 CA 3F D6 BB CA 1B D7 C3 54 D6 7B
D640 : 32 FA D1 DD 22 FB D1 3E 4F DD 77 00 3E 0C DD 77
D650 : 01 C3 FE D6 DD 34 0F DD 34 10 DD 34 11 DD 7E 07
D660 : DD BE 10 F2 83 D6 AF DD 77 10 CD 44 D3 DD 2A 8D
D670 : D1 3A FD D1 DD 86 00 DD 77 00 FE A0 D4 80 D3 DD
D680 : 2A 8D D1 DD 7E 08 DD BE 11 F2 A9 D6 AF DD 77 11
D690 : CD 44 D3 DD 2A 8D D1 3A FD D1 DD 86 01 DD 77 01
D6A0 : FE 4C F4 80 D3 DD 2A 8D D1 DD 7E 0B FE 10 F2 BC
D6B0 : D6 3E 10 DD 86 0B DD 77 0B DD 77 0D DD 7E 0B DD
D6C0 : BE 0F F2 E8 D6 AF DD 77 0F DD 7E 06 DD 86 0C DD
D6D0 : 77 06 A7 CC 80 D3 DD 2A 8D D1 DD 7E 06 FE 09 FA
D6E0 : E8 D6 CD 80 D5 CD 80 D3 DD 2A 8D D1 3A 02 D2 DD
D6F0 : 86 00 DD 77 00 3A 03 D2 DD 86 01 DD 77 01 CD F0
D700 : D3 DD 2A 8D D1 DD 7E 00 FE 55 CA 1B D7 F2 17 D7
D710 : 21 8A D1 34 C3 1B D7 21 8B D1 34 DD 7E 09 32 96
D720 : D1 DD 7E 0A 32 97 D1 CD 10 D3 CD F0 D2 3A A0 D1
D730 : 21 89 D1 86 77 7B FE 05 C2 FB D5 C9 00 00 00 00
D740 : 3E 0B 32 97 D1 3A 89 D1 A7 C2 9F D7 3E 05 32 96
D750 : D1 21 A4 D1 22 98 D1 3A 8A D1 A7 CA 69 D7 CD F0
D760 : D2 3E 01 32 04 D2 C3 77 D7 3A 04 D2 A7 CA 77 D7
D770 : CD 00 D3 AF 32 04 D2 3E 43 32 96 D1 21 BC D1 22
D780 : 98 D1 3A 8B D1 A7 CA 92 D7 CD F0 D2 3E 01 32 05
D790 : D2 C9 3A 05 D2 A7 C8 CD 00 D3 AF 32 05 D2 C9 3A
D7A0 : 04 D2 A7 CA B8 D7 3E 05 32 96 D1 21 A4 D1 22 98
D7B0 : D1 CD 00 D3 AF 32 04 D2 3A 05 D2 A7 C8 3E 43 32
D7C0 : 96 D1 21 BC D1 22 98 D1 CD 00 D3 AF 32 05 D2 C9
D7D0 : 00 00 00 00 00 00 00 00 3A 8F D1 21 60 D0 01 0A
D7E0 : 00 3D CA E9 D7 09 C3 E1 D7 22 94 D1 21 A1 D1 22
D7F0 : 98 D1 C9 00 00 00 00 00 DD 7E 04 A7 CA 07 D8 ED
D800 : 44 DD 77 04 C3 10 D8 CD 44 D3 3A FD D1 DD 77 04
D810 : 3A 01 D2 FE 01 C2 34 D8 3A FE D1 3C 47 DD 77 02
D820 : DD 7E 04 FE 01 C2 2E D8 3E 27 80 C3 41 D8 3E 27
D830 : 90 C3 41 D8 3A FF D1 06 06 80 DD 77 02 47 C3 20
```

マシン語プログラム

```
D840 : D8 DD 77 00 DD 7E 05 A7 CA 53 D8 ED 44 DD 77 05
D850 : C3 5C D8 CD 44 D3 3A FD D1 DD 77 05 3A FE D1 DD
D860 : 77 03 DD 7E 05 FE 01 C2 72 D8 3E 0C DD 86 03 C3
D870 : 77 D8 3E 0C DD 96 03 DD 77 01 AF DD 77 08 DD 77
D880 : 09 CD 44 D3 C9 00 00 00 DD 2A 94 D1 DD 7E 00 FE
D890 : 27 C2 9E D8 AF DD 77 02 DD 77 04 C3 BB D8 F2 B1
D8A0 : D8 3E FF DD 77 04 3E 27 DD 96 00 DD 77 02 C3 BB
D8B0 : D8 D6 27 DD 77 02 3E 01 DD 77 04 DD 7E 01 FE 0C
D8C0 : C2 CB D8 AF DD 77 03 DD 77 05 C9 F2 DC D8 3E FF
D8D0 : DD 77 05 3E 0C DD 96 01 DD 77 03 C9 D6 0C DD 77
D8E0 : 03 3E 01 DD 77 05 C9 00 00 00 00 00 00 00 00 00
D8F0 : CD 88 D6 3E 27 DD 96 02 CB 2F DD 77 06 3E 0C DD
D900 : 96 03 CB 2F DD 77 07 C9 3A 01 D2 32 91 D1 32 93
D910 : D1 ED 44 32 90 D1 32 92 D1 1E 00 1C 7B 32 8F D1
D920 : CD D8 D7 DD 2A 94 D1 DD 7E 00 32 96 D1 DD 7E 01
D930 : 32 97 D1 CD 00 D3 DD 2A 94 D1 DD 7E 02 A7 CA 68
D940 : D9 DD 34 08 DD 7E 06 DD BE 08 F2 68 D9 AF DD 77
D950 : 08 DD 7E 04 FE 01 F2 5F D9 3A 90 D1 C3 62 D9 3A
D960 : 91 D1 DD 86 00 DD 77 00 DD 7E 03 A7 CA 96 D9 DD
D970 : 34 09 DD 7E 07 DD BE 09 F2 96 D9 AF DD 77 09 DD
D980 : 7E 05 FE 01 F2 8D D9 3A 92 D1 C3 90 D9 3A 93 D1
D990 : DD 86 01 DD 77 01 3A 02 D2 DD 86 00 DD 77 00 3A
D9A0 : 03 D2 DD 86 01 DD 77 01 CD 68 D8 DD 2A 94 D1 DD
D9B0 : 7E 02 DD B6 03 CC F8 D7 DD 2A 94 D1 DD 7E 02 FE
D9C0 : 27 D4 F8 D7 DD 7E 03 FE 0B D4 F8 D7 DD 7E 00 32
D9D0 : 96 D1 DD 7E 01 32 97 D1 CD F0 D8 CD F0 D2 7B FE
D9E0 : 0A C2 1B D9 C9 3A E1 D1 A7 C8 21 E2 D1 35 7E 32
D9F0 : F7 D1 D6 0C 47 C6 27 4F 90 90 47 AF 32 F6 D1 C5
DA00 : CD 20 D2 C1 3A E1 D1 FE 01 C2 A0 DA 3E 20 D3 40
DA10 : 2A F8 D1 58 16 00 19 3E 5A 77 2A F8 D1 59 19 3E
DA20 : A5 77 3A FA D1 A7 CA 31 DA 3A E2 D1 FE 10 FA 54
DA30 : DA DD 2A F8 D1 3A E0 D1 DD 77 59 DD 77 55 3E 58
DA40 : DD 77 53 DD 77 57 DD 70 52 04 DD 70 54 DD 71 56
DA50 : 0C DD 71 58 3A E2 D1 FE 0D F0 3A FA D1 A7 CA 6F
DA60 : DA CD 70 D5 CD 10 DB 3A FA D1 32 68 D1 CD 10 D3
DA70 : CD 80 D3 AF 32 FA D1 47 3A 8C D1 4F 2A 06 D2 09
DA80 : 22 06 D2 2A 11 D2 23 22 11 D2 3E 20 32 10 D2 3E
DA90 : 02 32 E1 D1 3A E2 D1 FE 0D F0 3E 18 32 E2 D1 C9
DAA0 : 3A 10 D2 D3 40 2A F8 D1 16 00 58 AF 19 77 2A F8
DAB0 : D1 59 19 77 2A F8 D1 01 52 00 09 4F 3C 71 23 FE
DAC0 : 08 C2 BC DA 3A 10 D2 A7 CA DF DA CD 10 DB 3A D8
DAD0 : D1 A7 C2 DB DA CD 70 D5 C3 DE DA CD 60 D5 00 3A
DAE0 : FA D1 A7 C2 61 DA AF D3 40 3A E2 D1 FE 0D F0 AF
DAF0 : 32 E1 D1 32 10 D2 3E 22 32 96 D1 3E 0A 32 97 D1
DB00 : 21 00 DD 22 98 D1 CD 00 D3 CD 60 D5 C9 00 00 00
DB10 : 3E 22 32 96 D1 3E 0A 32 97 D1 21 00 DD 22 98 D1
DB20 : CD F0 D2 C9 00 00 00 00 00 00 00 00 00 00 00 00
DB30 : 21 E7 F6 7E EE A0 77 21 5F F7 7E EE A0 77 21 D7
DB40 : F7 7E EE A0 77 21 B7 F9 7E EE 50 77 21 2F FA 7E
```

```
DB50 : EE 50 77 21 A7 FA 7E EE 50 77 06 00 21 BC F8 04
DB60 : 23 7E EE 40 77 7D C6 10 6F 7E EE 04 77 7D D6 10
DB70 : 6F 78 FE 06 C2 5F DB CD 70 DD 2A D6 D1 23 22 D6
DB80 : D1 CB 6C C8 3E 01 6C 32 F4 D1 C9 00 00 00 00 00
DB90 : 00 00 00 00 00 00 00 00 00 00 00 00 00 DB 09 CB
DBA0 : 4F C2 A8 DB 32 F4 D1 C9 DB 08 CB 77 C2 B7 DB 3E
DBB0 : FF 32 01 D2 C3 BC DB 3E 01 32 01 D2 DB 05 CB 47
DBC0 : C2 D4 DB 3A E1 D1 A7 C2 D4 DB 3E 01 32 E1 D1 3E
DBD0 : 17 32 E2 D1 DB 00 06 00 CB 67 C2 DE DB 04 CB 77
DBE0 : C2 E4 DB 05 78 CB 27 32 02 D2 06 00 DB 00 CB 57
DBF0 : C2 F4 DB 05 DB 01 CB 47 C2 FC DB 04 78 32 03 D2
DC00 : 1E 00 1C 7B 32 88 D1 CD 10 D3 DD 2A 8D D1 3A 01
DC10 : D2 DD BE 0E C2 28 DC DD 7E 0D CB 2F DD 77 0B DD
DC20 : 7E 0E DD 77 0C C3 43 DC DD 46 0D 3E 1F 90 DD 77
DC30 : 0B 3A 01 D2 DD 77 0C DD 7E 06 FE 06 FA 43 DC AF
DC40 : DD 77 0C 7B FE 05 C2 02 DC C9 00 00 00 00 00 00
DC50 : 1E 00 1C 7B 32 88 D1 CD 10 D3 CD 80 D3 7B FE 05
DC60 : C2 52 DC AF 32 D7 D1 32 D9 D1 32 E1 D1 32 FA D1
DC70 : 32 06 D2 32 07 D2 32 0A D2 32 11 D2 32 0C D2 32
DC80 : 04 D2 32 05 D2 32 12 D2 CD C0 D4 3E 98 32 E0 D1
DC90 : 3E 0A 32 8C D1 CD 80 D4 CD 50 D4 C9 00 00 00 00
DCA0 : AF 32 F4 D1 CD E0 D3 CD 90 DB CD 08 D9 CD F0 D5
DCB0 : CD 40 D7 CD E5 D9 CD 30 DB 3A F4 D1 A7 CA A7 DC
DCC0 : C9 3A 01 D2 FE FF C0 3E 50 DD 77 01 C9 00 00 00
DCD0 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DCE0 : DD 21 96 D1 DD 4E 04 06 00 2A 08 D2 09 22 08 D2
DCF0 : C9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DD00 : 0B 05 00 04 01 84 00 E0 C0 02 40 00 04 00 02 B2
DD10 : FE CF FF FF FE 4E 05 80 00 10 D9 FF FF FF FF FF
DD20 : FF 82 00 00 22 26 B9 7F FF FF F7 23 40 08 20 10
DD30 : 00 00 00 0F 31 10 10 05 00 01 50 02 31 03 10 05
DD40 : 03 01 32 96 D1 3E 0A 32 97 D1 21 00 DD 22 02 50
DD50 : 46 4E 23 D5 56 3E 20 D3 40 15 C2 59 DD AF D3 40
DD60 : 0B 78 B1 C2 54 DD 23 D1 C9 00 00 00 00 00 00 00
DD70 : 2A D6 D1 3E 20 94 47 4D 21 19 F3 00 A7 CA 86 DD
DD80 : 3D 71 23 C3 7C DD 78 A7 00 0E 80 3C 71 23 FE 20
DD90 : FA 8B DD 3E 20 32 51 F3 32 91 FE 3E F8 32 53 F3
DDA0 : 32 93 FE 3E 01 32 52 F3 32 92 FE C9 FF FF FF FF
DDB0 : CD 80 D4 CD 50 D4 C3 A7 DC FF FF FF FF FF FF FF
DDC0 : 08 04 00 80 24 AA AA 42 08 00 E0 C1 92 EC CE 29
DDD0 : 1C 0E 70 38 94 73 37 49 83 07 00 10 42 55 55 24
DDE0 : 01 00 00 00 00 00 00 00 FD 2A 8D D1 FD 7E 06 FE
DDF0 : 08 C0 3A D6 D1 CB 47 C0 21 C0 DD 22 98 D1 C9 00
DE00 : FF
```



マシン語プログラム・チェックサム

```
D0C8 - D107 = 07FB
D108 - D147 = 0BED
D148 - D187 = 106F
D188 - D1C7 = 0F0D
D1C8 - D207 = 0F0E
D208 - D247 = 1653
D248 - D287 = 1FE0
D288 - D2C7 = 23DD
D2C8 - D307 = 1BEA
D308 - D347 = 1B8F
D348 - D387 = 197C
D388 - D3C7 = 210F
D3C8 - D407 = 1E33
D408 - D447 = 1C7B
D448 - D487 = 16E2
D488 - D4C7 = 1E23
D4C8 - D507 = 1E80
D508 - D547 = 25FD
D548 - D587 = 1B02
D588 - D5C7 = 237C
D5C8 - D607 = 2153
D608 - D647 = 24E3
D648 - D687 = 22D7
D688 - D6C7 = 24CD
D6C8 - D707 = 2308
D708 - D747 = 20B3
D748 - D787 = 221B
D788 - D7C7 = 204F
D7C8 - D807 = 1A80
D808 - D847 = 1F34
D848 - D887 = 2072
D888 - D8C7 = 2303
D8C8 - D907 = 1A32
D908 - D947 = 211B
D948 - D987 = 2334
D988 - D9C7 = 2387
D9C8 - DA07 = 263D
DA08 - DA47 = 21E0
DA48 - DA87 = 216E
DA88 - DAC7 = 1C9F
DAC8 - DB07 = 24F6
DB08 - DB47 = 1C18
DB48 - DB87 = 1F70
DB88 - DBC7 = 1BA2
DBC8 - DC07 = 1E86
DC08 - DC47 = 1EE8
DC48 - DC87 = 1BD8
DC88 - DCC7 = 2775
DCC8 - DD07 = 0A23
DD08 - DD47 = 16E3
DD48 - DD87 = 1A44
DD88 - DDC7 = 24B8
DDC8 - DE00 = 16B2
```

PC-8001+マシン語

3次元ギャラクシアン・ゲーム
STARTREK FIRE
スタートレック・ファイア

東大PCユーザーズグループ　雄城嘉史

『FIRE』シリーズ第2弾をお届けします．今回の目標は画面設定を含めすべてマシン語で行なうこと，画面上で動く主要な物体に1機ごとに固有の色をつけること，および移動する物体のコースを任意の直線に指定することなどでした．

画面上に固有の色を持つ物体を表示する方法，すなわちアトリビュート・エリアの制御法はすでに何人かの人が確立していますが，私は私なりの方法をトライしてみました．ゲームは『3次元ギャラクシアン』っぽく，かつスタートレックの映画のごとく，適当にデザインしたオリジナル・ゲームです．

『エンタープライズ号艦長殿．銀河系内各セクションに侵入したクリゴン戦艦を掃討せよ!!』

1 ゲームの説明

プログラムは&HC600から&HE466までのざっと8K弱の領域を使います．もちろん16Kシステムでも動きます．

ロードの仕方は**mon**＋ RETURN ＋ L ＋ RETURN です．ロードし終わったらRUNさせるわけですが，キーの説明を見たかったり，音の出る出ないを制御したかったら**GE318** RETURN と押してください．**写真1**のように表示されます．

ロード前の時点で画面幅およびスクロール設定がどうなっていようとE318番地をCALLした時点で**WIDTH36, 25 ; CONSOLE0, 25**に設定されます．ステータス・バッファを使わない画面の設定法の一例なので，リストのE318番地を参照してください．

“**イマハ　ヨル　デスカ？**”と尋ねてくるので，音を出して良ければ N と，静かにプレイしたければ Y と押してください．

もう1つのスタート番地はE200番地です．**GE200** RETURN とモニタ内で打つか，または先ほどの質問に答えることにより，画面は**WIDTH80,25 ; CONSOLE0,25,0,1 ; COLOR7,,1**の状態に入ります．これもステータス・バッファは使っていません．

“**STARTREK**”という文字が様々な色で1文字ずつゆっくりと表示され，“**FIRE!!**”という文字が赤でパッと出現し，背後に星が流れ続けます(**写真2**)．

なお，私の友人でPCのゲームの効果音はHAL研究所製のPCGなどをつながなければできないと思っていた者がおりますがそんなことはありません．このゲームも前作の『STAR FIRE』もプログラマブル・サウンド・ジェネレータなどは必要ありません．

次に，デモンストレーションとともに登録されているHIGH SCOREが表示されます(**写真3**)．このデモンストレーションはダミーのエンタープライズが破壊されるまで続くので，運が悪いとちょっと長くなるかもしれません．

続いてエンタープライズがワープし，“**SECTION00**”という文字が表示されます(**写真4**)．そして，ゲームの開始です．キーボードの説明を**図1**に示します．

このゲームは3次元ギャラクシアンといった風に作るために画面中央上方に消失点(無限遠点)を設けてあります．ゲームの目的は光子魚雷を持ってクリンゴン戦艦を破壊することですが，光子魚雷は消失点に向かい進行し，エンタープライズも消失点に向かい方向および形を変え，星も消失点より湧き出します．

したがって，画面内において光子魚雷が存在しうる範囲は**図2**に示す三角形の部分だけです．この領域内に敵艦を持ってくるために旋回するのです．旋回により星と敵艦は**押したキーと逆方向に**流れますが，エンタープライズの光子魚雷とクリンゴン側光子魚雷はコースを変えません．不合理ではありますが，これを変えると易しすぎたり難しすぎたりするので，クリンゴン側光子魚雷をよけるために平行移動があります(**写真5**)．

プレイ中の様子を**写真6**に示します．赤いのが敵光子魚雷です．1面消し終わるとエンタープライズはワープし，次のセクションに進みます．

写真 1　キーと得点の説明．昼か夜かの問い合わせもあります．

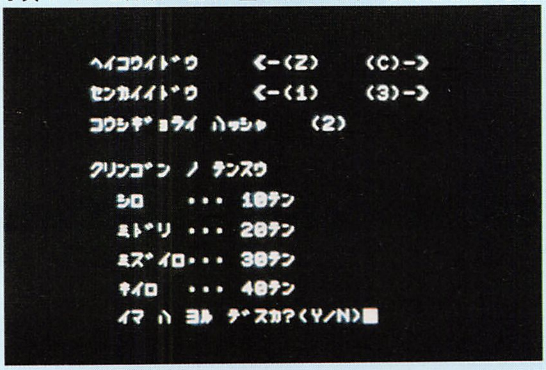


写真 2　タイトル文字の後を星が放射状に流れる．



写真 3　デモ画面．HIGH SCOREは 0 点．



写真 4　エンタープライズはSECTION 00へワープ．いよいよ戦いが…



写真 5　エンタープライズ号を左へ平行移動する．

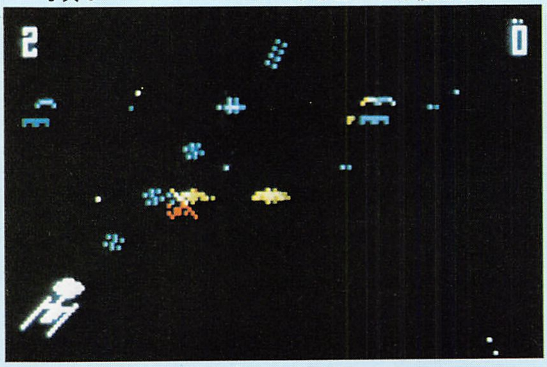


写真 6　敵の光子魚雷が接近！

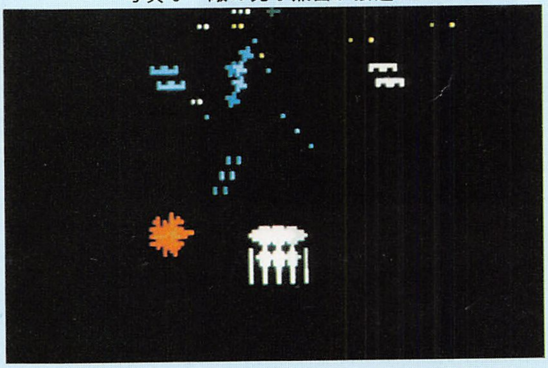


写真 7　GAME OVERの表示



写真 8　4,600点がHIGH SCOREに登場されたところ．



図1　キーボードの説明

- ●平行移動(エンタープライズの左右移動)　←Z　C→
- ●旋回移動(クリンゴン艦を照準内に)　←1　3→
  ただし，矢印は旋回方向であり，画面内の動きは逆．
- ●光子魚雷発射　2
- ●ゲーム中止(GAME OVERへ)　STOPキー
- ●モニタに戻る(GAME OVER時)　STOPキー
- ●リプレイ(GAME OVER時)　スペースバー

図2　エンタープライズの射線

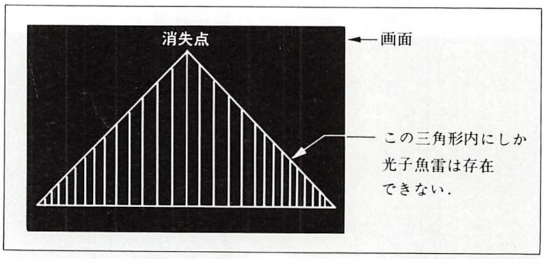


写真6で画面右側上方にあるのが点数，画面左側上方にあるのが残りのエンタープライズ機数です．なお，エンタープライズの残り台数は画面上にあるものを含みません．初期値は3台です．得点は999,999点まで表示できます．

敵の発砲頻度は面ごとに変わり，SECTION 11，27，43，……で最高になります．また，クリンゴン艦はその色によって得点が異なります(表1)．さて，エンタープライズはクリンゴン艦の体当たりを受けたり，敵光子魚雷を受けると爆発します．そして，3台すべてがやられるとGAME OVERになります(写真7)．

特に，得点が前回までに出されたHIGH SCOREを上回っていると，その旨表示し，自動登録します(写真8)．GAME OVERの状態でSTOPキーを押すとWIDTH36，25の状態になり，モニタに戻ります．

一方，SPACEバーを押すとリプレイできます．図1に示したようにSTOPキーはプレイ中止のためのキーで，プレイ中にSTOPキーを押すとGAME OVERの状態になるので，あまり長く押していると一気にモニタに戻ってしまいます．

なお，STOPキーでゲームが中止されるのはキー入力を受け付けるときだけなので，ワープやデモ中は中止できません．HIGH SCOREのクリア方法は&HCE75～&HCE78の4バイトをモニタから&H00に直すことで実行できます．

2　プログラムの説明

プログラムの占める領域&HC600から&HE466までのうち，データ・エリアが&HC600から&HCCFFまで，ワーキング・エリアが&HCD00から&HCEFFまでを占めます．

なぜデータ・エリアがこんなに大きくなったかと言うと，エンタープライズが14通り，クリンゴン艦が32通り，クリンゴン光子魚雷が16通り，エンタープライズ光子魚雷が8通り，他にグラフィックで描いた文字が27通りの画像を使っているためです．

表2と表3に主要サブルーチンの説明および主要ワーク・エリアの説明を示します．図3のフローチャートと共に参照してください．なお，プログラム自体は例によって破滅的なので，あまり見られない方が良いのでは……．

音出し，音消しでは&HDF06番地を直接書き替えています．この範囲のチェック・サムが狂っていたときにはまず，ここを見てください．

表1　クリンゴン艦1隻当たりの点数

色	点数
白	10
緑	20
水色	30
黄色	40

なお，クリンゴン艦は5段階に大きくなる．

表2　主要サブルーチンの説明

開始番地	名前	説明
CF00H	PRMODE	ゲーム開始時の画面設定および準備
CF60H	ENMODE	ゲーム終了時の画面設定
CF6DH	KLINUM	クリンゴン艦の状態を出力し，ワーク・エリアを頭出しする．
CF89H	KLMINU	同クリンゴン光子魚雷
CFA5H	ENMNUM	同エンタープライズ光子魚雷
CFC1H	STANUM	同星(画面内に10個存在する)
CFD2H	DATKLI	描こうとするクリンゴン艦の画像データを出力
D066H	DATKLM	同クリンゴン光子魚雷
D0ADH	DATENM	同エンタープライズ光子魚雷
D850H	DATENT	同エンタープライズ
DDB8H	WARDAT	同ワープ中エンタープライズ
D0E7H	KLMSCN	待機中のクリンゴン光子魚雷を探す．
D0FCH	ENMSCN	同エンタープライズ光子魚雷
D111H	PUT	画像を描くか消す．
D222H	ATTRIB	アトリビュート・エリア制御
D2F0H	DELCUL	移動物体にコースを与える．
D34CH	VECTCU	移動物体に動く方向を指示する．
D3F0H	SRND	ROMより乱数を発生
D4D0H	STACON	星の制御
D560H	AIMFIR	クリンゴン光子魚雷の照準
D5C0H	KLIRND	クリンゴン艦の初期化
D608H	KLFJUD	クリンゴン艦爆発を判断
D67CH	KLNOCN	生きているクリンゴン艦の制御
D70CH	KLMOVE	クリンゴン艦の移動
D980H	KLABCN	爆発中のクリンゴン艦を制御
D7ACH	ENFJUD	エンタープライズに命中したか否かの判断
D8D4H	KLMCON	クリンゴン光子魚雷制御
D9DCH	ENTMCN	エンタープライズ側光子魚雷制御
DA60H	ENMFIR	エンタープライズ光子魚雷発射
DAB0H	ENTCON	エンタープライズ制御
DAF8H	INPUT	キー入力
DE04H	WARP	ワープ制御
D7E8H	GMPREP	面が変わるごとの初期化
DE9CH	PREP	ゲーム開始時の初期化
DEF6H	MAIN	ゲーム部主ルーチン
E1E4H	DEMONS	ゲーム開始時のデモ制御
E200H	HMAIN	最上位ルーチン

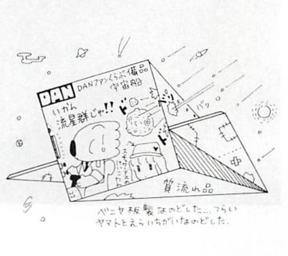
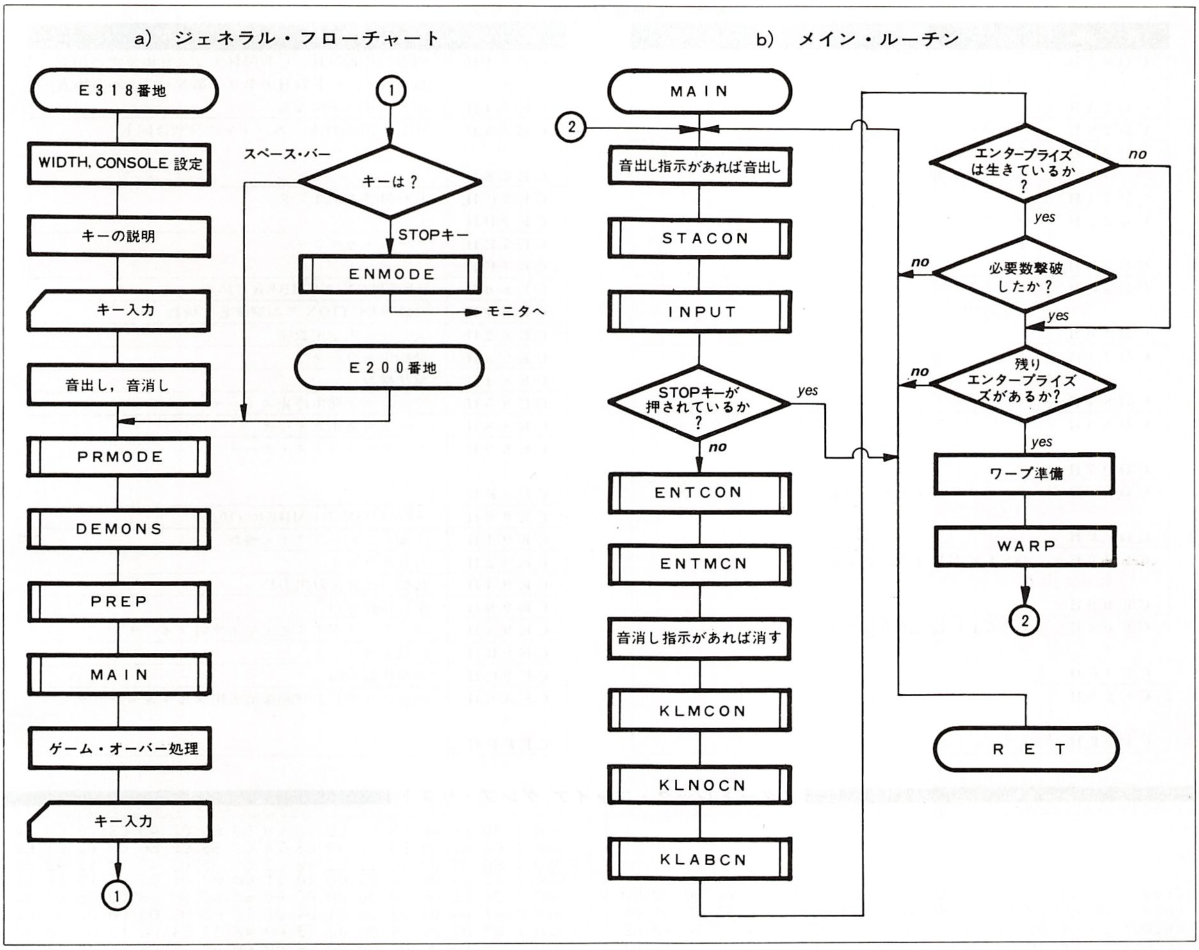


図３　フローチャート(概略)



3 変更点

変更可能な場所は２箇所だけです．１つは各ＳＥＣＴＩＯＮでのクリンゴン機数と発砲頻度で，これは＆ＨＣＣＥ８から＆ＨＣＣＦＦまでテーブルになっています．偶数バイトが各面機数，奇数バイトが発砲頻度です．発砲頻度は何ビット立っているかで変化し，立っているビット数が少ないと発砲頻度は増します．

第２は反射神経の特に優れている人のみにお勧めしますが，このゲームの速度を押さえているのは主としてアトリビュート・エリア制御なのです．だから『画面は真っ白でいい！とにかく限界まで速く！』という人には＆HD222番地に＆ＨＣ９を書いてみることをお勧めします．

いまのままでも相当な反射神経が必要ですが，ちなみに作者のアベレージは12,000点ほどです．

4 注意点

各面機数に関係なく，画面内に同時に存在するクリンゴン艦は５隻までです．同様にエンタープライズの光子魚雷数も５，クリンゴン側光子魚雷は10までです．

クリンゴン艦側の照準は正確にエンタープライズを狙ってくるものと，わざとエンタープライズの動こうとする方向にずらしてくるものと半々です．

5 最後に

アトリビュート・エリア制御に関するＲＯＭ内サブルーチンを読んで見るとＰＣはキャラクタ単位での色変更や属性変更を念頭に置いたマイコンであることがわかります．

このゲームの中で使用する画像は数10キャラクタより成る領域を占めるものが多く，そのため，ＲＯＭ内ルーチンによりキャラクタ単位で色を付けていたのでは遅過ぎるし，また，アルゴリズムの良い練習問題にもなるので，ＲＯＭ内サブルーチンはあまり使わず自分で色制御部を作ってみました．なお，一度色宣言をしたエリアはしばらくその色が残るので星にも色が付きます．

私としては宇宙ものはこれでおしまい，次回からは少し毛色の異なる『ファイア！』シリーズを作りたいと思います．

なお，このゲームはＲＯＭ内サブルーチンをかなり使っていますが，いまのところ，どのバージョンでも大丈夫のはずです（画面反転**OUT81,33**は使っていないので）．とにかく荘絶無双混乱極限のプログラムですが，お許しを．

表3　主要なワーク・エリア

番　地	説　　明
CD00H 〜 CD2AH	PUTおよびATTRIBの入出力およびスタック
CD2BH	NUMBER入力
CD2DH 〜 CD31H	クリンゴン艦状態表　00Hでノーマル，01Hで爆発，それ以外でパス
CD32H 〜 CD3BH	同クリンゴン光子魚雷　00Hで待機，01Hで進行中
CD3CH 〜 CD40H	同エンタープライズ光子魚雷　00Hで待機，01Hで進行中
CD41H 〜 CD87H	クリンゴン艦ワーク・エリア　各13バイト（余りあり）
CD88H 〜 CDD7H	クリンゴン光子魚雷ワーク・エリア　各8バイト
CDD8H 〜 CDFFH	エンタープライズ光子魚雷ワーク・エリア　各8バイト
CE00H 〜 CE09H	DELCUL入出力およびスタック
CE0AH 〜 CE1FH	VECTCU入出力およびスタック
CE20H 〜 CE6FH	星ワーク・エリア　各8バイト

番　地	説　　明
CE71H 〜 CE74H	得点　CE71H〜CE73Hまでに6桁までの10進数が入り，CE74Hの第0〜第5ビットが各桁表示のスイッチになる．
CE75H 〜 CE78H	HIGH SCORE　各バイトの役割は同上
CE7CH CE7DH	ROM読みポインタ
CE7EH CE7FH	ゲーム・カウンタ
CE80H	SECTION NUMBER（16進）
CE81H	当該SECTIONでの撃墜必要機数
CE82H	発砲頻度決定用数値
CE83H	発砲用カウンタ
CE84H	撃墜機数
CE85H	クリンゴン発生停止スイッチ
CE88H	キー入力禁止スイッチ
CE89H 〜 CE8FH	エンタープライズ・データ
CE90H	SECTION NUMBER（10進）
CE91H	予備エンタープライズ機数
CE92H	音長カウンタ
CE94H	乱数（SRND出力）
CE99H	平行移動方向
CE9AH	エンタープライズ光子魚雷発射スイッチ
CE9BH	同禁止カウンタ
CE9CH	旋回移動方向
CEA0H 〜 CEFFH	画面クリアおよび画像消去用ヌル・キャラクタ

スタートレック・ファイア ダンプ・リスト

```
                    DUMP LIST

C600 : 0A 04 00 00 00 00 00 40 EE FF EF 4E 00 00 00 60
C610 : AC 88 2C 11 81 00 00 80 24 81 EC 77 93 35 01 00
C620 : 20 01 00 00 80 24 01 00 00 00 07 04 00 00 40 EE
C630 : FF EF 4E 00 00 E8 88 9F 01 C0 00 68 81 7E 33 D1
C640 : 03 16 00 00 00 48 03 00 07 04 00 00 40 EE FF EF
C650 : 4E 00 00 68 80 BD 08 0C 00 68 81 FF 37 3C 00 60
C660 : 01 10 00 3C 00 00 06 04 00 40 EE FF EF 4E 00 78
C670 : 00 F9 09 E0 C0 03 E0 7F 83 07 16 00 13 00 3C 00
C680 : 05 04 E4 FE FF EE 04 0C 90 9F 00 0C 0F 71 7F 01
C690 : 0F 0F 00 00 00 0F 06 04 E4 FE FF EE 04 00 0E 90
C6A0 : 9F 00 87 00 70 38 F7 0E 30 0C 00 C3 00 31 00 61
C6B0 : 07 04 E4 FE FF EE 04 00 00 C0 80 DB 08 86 00 00
C6C0 : 00 C3 73 FF 18 86 00 00 00 C3 00 01 10 06 07 04
C6D0 : E4 FE FF EE 04 00 00 0C 10 F9 88 8E 00 00 30 1D
C6E0 : 33 E7 18 86 00 00 30 84 00 00 00 61 0A 04 E4 FE
C6F0 : FF EE 04 00 00 00 00 00 00 18 11 C2 68 CA 08 00
C700 : 00 00 00 10 53 39 77 CE 18 42 08 00 00 00 00 10
C710 : 42 08 00 00 10 02 00 2A 48 66 80 96 B0 CE EC 07
C720 : 05 04 03 02 02 02 02 02 0A 06 00 00 00 40 00 10
C730 : 80 00 00 00 00 00 01 80 41 10 04 00 82 00 00 00
C740 : 01 80 41 10 04 00 82 00 00 02 40 00 0A 05 14 82
C750 : 00 04 10 40 81 52 18 1C 1A 04 14 04 00 21 40 09
C760 : F5 37 27 14 28 00 10 17 1E 24 2A 30 37 3E 50 04
C770 : 04 E4 FE EE 04 8E FC 8C 0E 0F 73 03 0F 01 00 00
C780 : 01 03 03 66 FF 66 4F EE F4 07 00 07 03 03 7A 7F
C790 : 0A 2F 27 0F 01 00 01 02 02 7A 0A 17 07 6F 81 8C
C7A0 : 97 00 11 0D 09 00 07 02 4C 26 76 7F 26 46 0C 00
C7B0 : 00 10 13 00 00 00 03 04 00 71 0C E4 F4 EF 00 D0
C7C0 : 07 00 11 00 07 02 00 00 00 C8 08 00 00 30 62 64
C7D0 : FE 6E 64 32 03 04 00 88 00 E0 0B 00 F7 2F 27 30
C7E0 : 87 00 A6 B6 C4 D4 04 02 04 02 01 02 01 02 B0 6C
C7F0 : C1 8E D2 E3 8E 7D 00 00 0A 10 16 1C 2C 32 38 3E
C800 : FF BF 33 11 00 00 00 00 55 55 00 00 00 00 06 02
C810 : 88 C4 EE CE 84 08 00 00 32 02 00 00 03 03 00 B0
C820 : 0C 70 F2 7F 00 60 01 00 06 02 00 00 40 4C 00 00
C830 : 10 21 73 77 23 11 03 03 00 68 00 E0 FF E4 00 D3
C840 : 00 0E 1C 28 36 03 02 03 02 01 02 01 02 05 01 64
C850 : 62 6F 62 04 02 03 B0 0E B1 1F 10 00 05 01 20 46
C860 : F6 46 26 00 02 03 00 08 F8 8D 70 0D 4D 54 5C 64
C870 : 03 01 03 01 01 02 00 02 04 01 26 62 26 62 02 02
C880 : 90 0F 90 0F 04 01 46 64 46 64 02 02 F0 09 F0 09
C890 : 78 7E 84 8A 02 01 02 01 01 01 01 01 03 01 12 11
C8A0 : 21 01 02 E1 12 03 01 84 88 48 01 02 1E 21 9C A1
C8B0 : A5 AA 01 01 01 01 01 01 01 01 E2 C7 41 C8 6C C8
```

```
C8C0 : 90 C8 AE C8 C7 C8 C8 C8 C8 E6 C7 45 C8 70 C8 94
C8D0 : C8 B2 C8 EA C7 49 C8 74 C8 98 C8 B6 C8 02 01 E0
C8E0 : EA 03 02 2C 11 C2 10 22 01 05 03 48 12 11 42 08
C8F0 : 43 08 00 48 03 00 10 11 00 00 06 03 6C 13 11 11
C900 : 31 C6 0F 00 00 00 00 F0 63 8C 88 88 C8 36 03 02
C910 : A1 A4 01 01 00 01 04 02 88 F8 88 08 00 70 00 00
C920 : 05 03 21 84 00 48 12 00 48 33 84 00 12 00 00 00
C930 : 21 06 03 00 00 00 0F 00 00 44 44 44 4F 44 44 00
C940 : 00 00 0F 00 00 02 01 E2 47 03 02 90 02 24 01 10
C950 : 02 05 03 01 02 84 00 14 80 12 84 12 00 02 00 00
C960 : 10 20 06 03 20 40 28 04 80 02 40 82 14 28 41 02
C970 : 40 01 20 14 02 04 E1 C8 E9 C8 FA C8 0E C9 16 C9
C980 : 20 C9 31 C9 45 C9 49 C9 51 C9 62 C9 01 01 02 03
C990 : 01 02 02 03 01 01 02 03 07 08 0B 0C 01 02 03 04
C9A0 : 05 01 06 02 07 03 08 04 09 0A 01 02 05 06 09 01
C9B0 : 01 02 09 0A 05 06 07 01 01 02 01 09 0A 09 05 01
C9C0 : 98 A0 A8 B0 B8 02 01 E2 02 01 E2 47 02 01 14 28
C9D0 : 02 01 F4 44 02 01 82 41 02 01 44 4F 02 01 18 81
C9E0 : 02 01 1D B8 02 01 06 60 C8 CC D0 D4 D8 DC E0 E4
C9F0 : 00 04 00 08 0C 16 20 2A 34 3E 48 52 5C 70 84 98
CA00 : 02 01 96 69 02 01 09 90 02 01 60 06 04 02 00 80
CA10 : 0C 08 00 62 F3 15 04 02 04 40 40 00 0A 3B A0 00
CA20 : 04 02 40 CE 4E 00 10 13 13 00 04 02 00 49 21 58
CA30 : 00 02 21 20 04 02 41 FC 7E 14 15 F5 B1 51 04 02
CA40 : 08 82 28 08 10 14 41 10 04 02 21 84 48 12 48 12
CA50 : 21 84 04 02 20 28 04 0A 50 20 04 04 06 03 00 00
CA60 : F0 00 00 00 44 44 E2 47 22 22 00 00 00 0F 00 00
CA70 : 06 03 00 E4 C0 E8 6C 04 50 F7 FF FF 7F 45 10 31
CA80 : F1 73 31 01 06 03 40 82 4D D4 28 04 09 9F 60 06
CA90 : F9 90 20 14 2B B2 41 02 06 03 28 40 41 A0 50 40
CAA0 : A4 41 41 4A 0A 29 20 29 94 24 84 02 1B 02 8F 88
CAB0 : 0F F1 01 1F 11 07 8F 88 0F 00 F0 99 B9 F0 11 31
CAC0 : F0 11 F1 F0 99 F9 F0 99 19 0F 00 0F F8 08 8F AA
CAD0 : 0E 0F 00 0F 00 C0 88 F8 F0 88 C8 F0 88 F8 F0 31
CAE0 : 84 F0 88 88 14 02 9F 99 0B 9F 19 01 1F 11 03 11
CAF0 : 1F 01 F1 01 1F 11 0F 2F 84 0F 8C 88 0F 8F 88 08
CB00 : 8F 88 0C 00 0F 00 F8 08 8F 88 0F 0F 00 0F 02 02
CB10 : 1F F1 8F F8 02 02 10 0F 80 8F 02 02 9B F9 8F C8
CB20 : 02 02 93 F9 8C F8 02 02 0F 0F 11 1F 02 02 9F 99
CB30 : 8C F8 02 02 9F B9 8F F8 02 02 17 F1 00 F0 02 02
CB40 : 9F F9 8F F8 02 02 9F F9 88 F8 05 03 FF 33 33 33
CB50 : 77 77 66 66 66 EE EE CC CC CC FF 05 03 77 33 FF
CB60 : 33 77 00 00 FF 00 00 00 C0 FF 0C 00 05 03 FE 33
CB70 : 33 33 EF FF 33 33 33 FF FF 0C 00 C0 FF 05 03 FE
CB80 : 33 33 33 EF FF 73 FF 3B 13 FF 0C 10 F3 CE 05 03
CB90 : FF 33 33 33 77 FF 66 F6 0F 00 FF CC CC CC EE 05
CBA0 : 03 FF 80 EC 7F 03 FF F7 9F 00 00 FF 0C 31 E7 CC
CBB0 : 05 03 FF 33 33 33 FF FF 00 00 CC DD FF CC CC CC
```

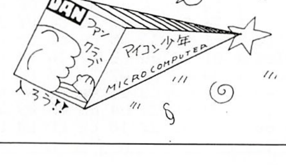
```
CBC0 : FF 05 03 FF CE 88 EC FF FF 10 33 01 FF FF 00 00
CBD0 : 00 FF 05 03 FF 33 33 33 FF FF 00 00 00 FF FF CC
CBE0 : CC CC FF 05 03 FF 00 00 00 FF FF 08 00 80 FF 10
CBF0 : 73 EE 37 01 0E 14 1A 20 26 2C 32 38 3E 44 4A 5B
CC00 : 6C 7D 5B 7D 8E 9F 0A 11 18 1F 28 2F 36 3D 08 05
CC10 : 77 FF 7F 77 77 77 FF 0F 00 FF 8F 88 CC 0C 00 00
CC20 : 00 FF 3F 33 77 07 00 00 00 FF 0F 00 00 00 00 00
CC30 : E0 FF EF 0E 00 00 00 00 04 05 70 FF 7F 00 00 FF
CC40 : 0F 00 00 FF 0F 00 00 FF 0F 00 EE FF EF 0E 08 05
CC50 : F7 FF 77 77 77 F7 EF 00 F0 FF CC CC CC FC FF 00
CC60 : F0 FF F3 FF 19 11 01 00 F0 FF 00 31 F7 CE 08 00
CC70 : FE FF 0E 00 00 F1 FF EE 07 05 FF 7F 77 77 77 FF
CC80 : 0F FF 0F 00 80 88 00 00 FF 7F 77 F7 FF 00 00 FF
CC90 : 0F 00 00 00 00 00 FF EF EE EE EE FF 0F 05 05 FF
CCA0 : 0F 00 F0 FF FF 0F 00 F0 FF FF 0F 00 F0 FF 11 01
CCB0 : 00 10 11 EE 0E 00 E0 EE 0E 38 4E 78 9D 0B 19 20
CCC0 : 2E 3A 41 4E 5B 68 75 88 90 98 A0 A8 B0 B8 C0 C8
CCD0 : D0 D8 E0 E8 F0 F8 20 28 30 38 40 48 50 58 60 68
CCE0 : F8 10 98 20 B8 30 D8 40 1C FC 1E FC 20 F8 22 F8
CCF0 : 24 F8 26 F8 28 0F 2A 0F 2C 0F 30 0F 38 07 40 07
```

CD00～CEFFの間はワーキング・エリアです.

```
CF00 : 3E 80 D3 51 3E 01 32 61 EA 01 FF 00 CD F7 08 01
CF10 : 19 50 CD 3A 09 3E F8 32 5B EA CD 87 04 3E F8 32
CF20 : 09 FF 3E 0C CD 57 02 06 50 21 A0 CE AF 77 05 23
CF30 : C2 2D CF C9 AF 00 F5 CD 64 06 01 50 00 21 A0 CE
CF40 : ED B0 F1 3C FE 19 FA 35 CF C9 AF 00 F5 CD 5B 06
CF50 : 01 27 00 21 08 FF ED B0 F1 3C FE 19 FA 4B CF C9
CF60 : 01 FF 00 CD F7 08 01 19 24 CD 3A 09 C9 06 00 3A
CF70 : 2B CD 4F 21 2C CD 09 7E 32 2C CD 22 F4 CE 21 C1
CF80 : CC 09 4E 06 CD C5 DD E1 C9 06 00 3A 2B CD 4F 21
CF90 : 31 CD 09 7E 32 2C CD 22 F4 CE 21 C6 CC 09 4E 06
CFA0 : CD C5 DD E1 C9 06 00 3A 2B CD 4F 21 3B CD 09 7E
CFB0 : 32 2C CD 22 F4 CE 21 D0 CC 09 4E 06 CD C5 FD E1
CFC0 : C9 06 00 3A 2B CD 4F 21 D5 CC 09 4E 06 CE C5 DD
CFD0 : E1 C9 DD 7E 00 32 00 CD DD 7E 05 32 04 CD DD 7E
CFE0 : 02 32 02 CD DD 7E 01 32 01 CD 0E 00 FE 10 F2 04
CFF0 : D0 0C FE 0D F2 04 D0 0C FE 0A F2 04 D0 0C FE 07
D000 : F2 04 D0 0C 06 00 DD 7E 04 A7 C2 4E D0 21 C4 C8
D010 : 09 7E CB 21 21 BA C8 09 4E 23 46 26 00 DD 6E 02
D020 : 09 4E 47 ED 43 08 CD 0A C5 32 0A CD CB 3F 4F 3A
D030 : 00 CD 91 32 05 CD 81 81 3C 3C 32 12 CD E1 23 7E
D040 : 32 0B CD CB 3F 4F 3A 01 CD 91 32 06 CD C9 21 C0
D050 : C9 09 6E 26 C9 DD 4E 02 09 4E CB 21 21 74 C9 09
D060 : 4E 23 46 C3 23 D0 DD 7E 00 32 00 CD DD 7E 02 32
D070 : 02 CD 4F 06 00 DD 7E 01 32 01 CD FE 08 F2 84 D0
D080 : AF C3 9A D0 FE 0C F2 8E D0 3E 04 C3 9A D0 FE 10
D090 : F2 98 D0 3E 08 C3 9A D0 3E 0C 81 00 00 4F 21 F0
D0A0 : C9 09 4E 06 CA 3E 58 32 04 CD C3 23 D0 FD 7E 00
D0B0 : 32 00 CD FD 7E 01 32 01 CD FD 7E 02 32 02 CD 4F
D0C0 : 06 00 21 E8 C9 09 4E 06 C9 3E B8 32 04 CD C3 23
D0D0 : D0 01 01 05 C5 79 32 2B CD CD 6D CF FE 02 C1 C0
D0E0 : 0C 05 C2 D4 D0 37 C9 01 01 0A C5 79 32 2B CD CD
D0F0 : 89 CF A7 C1 C8 0C 05 C2 EA D0 37 C9 01 01 05 C5
D100 : 79 32 2B CD CD A5 CF A7 C1 C8 0C 05 C2 FF D0 37
D110 : C9 AF 32 07 CD 3A 05 CD FE 01 F8 FE 4F D0 3A 12
D120 : CD FE 4F D0 3A 06 CD FE 01 F8 4F FE 17 D0 3A 0B
D130 : CD 81 FE 01 F8 FE 17 D0 3A 06 CD 32 1E CD 3A 0A
D140 : CD 2A 08 CD 23 23 4F 06 00 ED 43 10 CD 3A 0B CD
D150 : F5 E5 3A 05 CD 67 24 3A 1E CD 3C 32 1E CD 6F CD
D160 : EA D2 E1 3A 03 CD A7 C2 58 E4 00 00 00 ED 4B 10
D170 : CD ED B0 F1 3D C2 50 D1 3E 01 32 07 CD 3A 03 CD
D180 : A7 CC 22 D2 C9 FF FF FF CD E4 DB CB 27 4F 06 00
D190 : 21 27 00 A7 ED 42 22 19 CD 2A 13 CD 01 27 00 09
D1A0 : 22 15 CD 2B 2B 22 17 CD ED 5B 15 CD ED 4B 19 CD
D1B0 : ED B8 C9 FF CD E4 DB CB 27 4F 06 00 21 27 00 A7
D1C0 : ED 42 22 19 CD 2A 13 CD 09 22 17 CD CD F1 DB CB
D1D0 : 27 4F 2A 13 CD 09 22 15 CD ED 5B 15 CD 2A 17 CD
D1E0 : ED 4B 19 CD 06 00 ED B0 C9 00 00 00 3A 1E CD 6F
D1F0 : 3A 24 CD 67 CD 25 06 AF 32 1F CD 3A 24 CD B9 C2
D200 : 11 D2 3E 01 32 1F CD 01 04 00 A7 ED 42 4E 23 46
D210 : 23 ED 5B 13 CD 22 1B CD A7 ED 52 CB 3D 7D CD F4
D220 : DB C9 3A 06 CD 32 1E CD 3A 0B CD 00 F5 3A 1E CD
D230 : CD 5B 06 ED 53 13 CD 3A 05 CD 32 24 CD CD EC D1
D240 : 3A 1F CD A7 CA 50 D2 3A 1D CD 32 27 CD CD 88 D1
D250 : CD 00 DC 3A 05 CD 77 23 3A 04 CD 77 78 32 98 CE
D260 : 23 22 25 CD 3A 1D CD 3C 32 29 CD 3A 12 CD 32 24
D270 : CD CD EC D1 78 32 98 CE 3A 29 CD 32 28 CD 3A 1D
D280 : CD 32 27 CD CD B4 D1 3A 1F CD A7 C2 A4 D2 3A 29
D290 : CD 32 27 CD CD 88 D1 3A 12 CD 2A 25 CD 77 23 3A
D2A0 : 98 CE 77 00 3A 1E CD 3C 32 1E CD F1 3D C2 2C D2
D2B0 : C9 00 00 00 2A 20 CD 3A 12 CD 77 23 3A 98 CE 77
D2C0 : C3 DD D2 3A 1D CD 32 27 CD 3A 29 CD 3C 32 28 CD
D2D0 : CD B4 D1 3A 1F CD A7 C2 DD D2 C3 B4 D2 3A 1E CD
D2E0 : 3C 32 1E CD F1 3D C2 45 D2 C9 CD F3 03 EB C9 00
D2F0 : 3A 02 CE 47 3A 00 CE 90 32 04 CE E6 80 CA 05 D3
D300 : 3E FF C3 07 D3 3E 01 32 07 CE 3A 03 CE 47 3A 01
D310 : CE 90 32 05 CE E6 80 CA 1F D3 3E FF C3 21 D3 3E
D320 : 01 32 06 CE A7 3A 04 CE F2 2D D3 ED 44 32 08 CE
D330 : 3A 07 CE A7 3A 05 CE F2 3C D3 ED 44 CD 93 DB C9
D340 : CB 78 C8 21 00 00 A7 ED 42 C3 A0 D4 2A 0A CE 06
D350 : 00 E5 CD D4 DB 4F 09 22 0E CE E1 CD DC DB 4F A7
D360 : ED 42 22 10 CE CD D4 DB 4F 09 22 12 CE ED 4B 0E
D370 : CE CD 40 D3 ED 43 14 CE ED 4B 10 CE CD 40 D3 ED
D380 : 43 16 CE ED 4B 12 CE CD 40 D3 ED 43 18 CE 3E 01
D390 : 32 1E CE 2A 0E CE 22 1A CE 2A 14 CE 22 1C CE ED
D3A0 : 4B 16 CE ED 42 FA B9 D3 3E 10 32 1E CE 2A 10 CE
D3B0 : 22 1A CE 2A 16 CE 22 1C CE 2A 1C CE ED 4B 18 CE
D3C0 : A7 ED 42 FA DA D3 CA DA D3 3E 11 32 1E CE 2A 12
D3D0 : CE 22 1A CE 2A 18 CE 22 1C CE 3A 1E CE E6 0F 32
D3E0 : 1F CE 3A 1E CE 0F 0F 0F 0F E6 0F 32 1E CE C9 00
D3F0 : 2A 7C CE ED 4B 7E CE 09 E5 01 FF 5F ED 42 E1 FA
D400 : 12 D4 CD 02 16 3A 76 EA 6F 3A 77 EA E6 03 67 22
D410 : 7E CE 7E 32 94 CE 4F 06 00 09 22 7C CE E6 3F 32
D420 : 95 CE E6 1F 32 96 CE E6 02 3D 32 97 CE C9 FF FF
D430 : AF 32 07 CD DD 66 00 7C 24 FE 01 F8 FE 4F F0 DD
D440 : 6E 01 7D 2C A7 F8 FE 18 D0 CD F3 03 3A 03 CD A7
D450 : CA 5E D4 7E FE 04 C2 66 D4 AF 77 C3 66 D4 7E A7
D460 : C3 48 E4 3E 04 77 3E 01 32 07 CD C9 3E 27 32 02
D470 : CE 3E 03 32 03 CE DD 7E 00 32 00 CE DD 7E 01 32
D480 : 01 CE CD F0 D2 2A 08 CE DD 75 04 DD 74 05 2A 06
D490 : CE DD 75 06 DD 74 07 AF DD 77 02 DD 77 03 C9 00
D4A0 : A7 E5 C1 C9 DD 77 01 CD 6C D4 C9 D5 CD 08 D6 D1
D4B0 : C9 00 00 00 CD 24 D8 3A 94 CE E6 1F C6 17 DD 77
D4C0 : 00 CD 24 D8 3A 94 CE E6 07 C6 02 CD A4 D4 C9 00
D4D0 : 2A 7E CE 23 22 7E CE 1E 0A D5 7B 32 2B CD CD C1
D4E0 : CF 3E 01 32 03 CD CD 30 D4 D1 D5 CB 43 C2 F8 D4
D4F0 : 3A 7E CE CB 47 C2 04 D5 CD 55 D5 3A 9C CE DD 86
D500 : 00 DD 77 00 AF 32 03 CD CD 30 D4 3A 07 CD A7 CC
D510 : B4 D4 D1 1D C2 D9 D4 C9 DD 6E 02 DD 66 03 22 0A
D520 : CE DD 6E 04 DD 66 05 22 0C CE CD 4C D3 2A 1A CE
D530 : DD 75 02 DD 74 03 3A 1E CE A7 CA 46 D5 DD 7E 07
D540 : DD 86 00 DD 77 00 3A 1F CE A7 C8 DD 7E 06 DD 86
D550 : 01 DD 77 01 C9 3A 9C CE A7 C4 6C D4 CD 18 D5 C9
D560 : CD 24 D8 3A 94 CE 4F 3A 83 CE 81 32 83 CE 4F 3A
D570 : 82 CE A1 C0 DD 6E 00 DD 66 01 22 02 CE DD E5 CD
D580 : E7 D0 D2 88 D5 DD E1 C9 3E 01 2A F4 CE 77 2A 02
D590 : CE DD 75 00 DD 74 01 AF DD 77 02 CD 06 E4 22 00
D5A0 : CE CD F0 D2 3A 07 CE DD 77 07 3A 08 CE DD 77 05
D5B0 : 3A 09 CE DD 77 06 AF DD 77 03 DD 77 04 DD E1 C9
D5C0 : 3A 85 CE A7 CA CE D5 2A F4 CE 3E 02 77 C9 CD 24
D5D0 : D8 3A 95 CE C6 08 DD 77 00 3E 04 DD 77 01 AF 2A
D5E0 : F4 CE 77 DD 77 02 DD 77 03 DD 77 04 CD 24 D8 3A
D5F0 : 94 CE E6 03 CB 27 4F 06 00 21 E0 CC 09 7E DD 77
D600 : 05 23 7E DD 77 0A C9 FF 2A F4 CE E5 1E 05 D5 7B
D610 : 32 2B CD CD A5 CF 3A 2C CD FE 01 C2 54 D6 3A 05
D620 : CD FD BE 00 CA 2A D6 D2 54 D6 3A 12 CD FD BE 00
D630 : DA 54 D6 3A 06 CD FD 96 01 CA 3F D6 D2 54 D6 47
D640 : 3A 0B CD B8 DA 54 D6 CD 5D D6 2A F4 CE AF 77 3C
D650 : D1 E1 77 C9 D1 1D C2 0E D6 E1 C3 30 D8 DD 46 0A
D660 : 3A 71 CE 80 27 32 71 CE 3A 72 CE CE 00 27 32 72
D670 : CE 3A 73 CE CE 00 27 CD 64 DB C9 00 1E 05 D5 7B
D680 : 32 2B CD CD 6D CF 3A 2C CD A7 C2 03 D7 3E 01 32
D690 : 03 CD CD D2 CF CD 08 D6 CD 11 D1 2A F4 CE 7E A7
D6A0 : C2 03 D7 3A 9C CE DD 86 00 DD 77 00 DD 7E 03 A7
D6B0 : CA BF D6 CD 3D D8 00 E6 03 DD 77 02 DD 77 03 CD
D6C0 : 24 D8 3A 94 CE E6 FE C2 CF D6 3E 01 DD 77 03 CD
D6D0 : 34 D8 DD 7E 01 FE 10 F2 E5 D6 CD 24 D8 3A 94 CE
D6E0 : E6 0F CC 2A D8 CD D2 CF CD 08 D6 CD AC D7 2A F4
D6F0 : CE 7E A7 C2 03 D7 32 03 CD CD 11 D1 3A 07 CD A7
D700 : CC CE D5 D1 1D C2 7E D6 C9 00 00 00 DD 7E 01 FE
D710 : 07 F2 19 D7 3C DD 77 01 C9 FE 0C F2 3A D7 CD F0
D720 : D3 3A 94 CE E6 02 3D DD 86 00 DD 77 00 CD F0 D3
D730 : 3A 97 CE DD 86 01 DD 77 01 C9 C2 68 D7 2A 8C CE
D740 : 22 00 CE DD 6E 00 DD 66 01 22 02 CE CD F0 D2 2A
D750 : 06 CE DD 75 0B DD 74 0C 2A 08 CE DD 75 08 DD 74
D760 : 09 AF DD 77 06 DD 77 07 DD 6E 06 DD 66 07 22 0A
D770 : CE DD 7E 08 32 0C CE DD 7E 09 32 0D CE CD 4C D3
D780 : 2A 1A CE DD 75 06 DD 74 07 3A 1E CE A7 CA 99 D7
D790 : DD 7E 0C DD 86 00 DD 77 00 3A 1F CE A7 C8 DD 7E
D7A0 : 0B DD 86 01 DD 77 01 C9 FF FF FF FF 3A 8F CE A7
D7B0 : C0 3A 0B CD 4F 3A 06 CD 81 FE 16 F8 3A 05 CD 4F
D7C0 : 3A 8C CE B9 F8 F5 3A 12 CD 4F F1 B9 CA D0 D7 D0
D7D0 : 3E 01 32 88 CE 32 8F CE 3E 18 32 92 CE 2A F4 CE
D7E0 : AF 77 C9 00 00 00 00 00 3E 27 CD A0 DB 3E 14 32
D7F0 : 8D CE AF 32 8F CE 32 88 CE 32 84 CE 32 2D CD 21
D800 : 00 00 22 2E CD 22 30 CD 22 85 CE 22 99 CE CD B0
D810 : DF 1E 05 D5 7B 32 2B CD CD 6D CF CD C0 D5 D1 1D
D820 : C2 13 D8 C9 D5 CD F0 D3 D1 C9 D5 CD 58 D9 D1 C9
D830 : 22 F4 CE C9 3A 7E CE E6 01 C0 C3 0C D7 3A 7E CE
D840 : E6 04 C2 4C D8 DD 7E 02 3C DD 77 02 DD 7E 02 C9
D850 : 06 00 48 21 66 C7 3A 8C CE BE DA 62 D8 23 0C C3
D860 : 59 D8 C5 79 32 8E CE 21 16 C7 09 4E 06 C6 ED 43
D870 : 08 CD C1 21 1F C7 09 3A 8C CE 32 00 CD 96 2A 08
D880 : CD 32 05 CD 86 3C 32 12 CD 3E 12 32 06 CD 3E 04
D890 : 32 0B CD 3E 14 32 01 CD 3E F8 32 04 CD 7E 32 0A
D8A0 : CD C9 FF FF 21 28 C7 22 08 CD 2A 8C CE 22 8D CE
D8B0 : 3E 58 32 04 CD 3E 10 32 06 CD 3E 06 32 0B CD 3E
D8C0 : 0A 32 0A CD 3A 8C CE D6 05 32 05 CD C6 0B 32 12
D8D0 : CD C9 00 00 1E 0A D5 7B 32 2B CD CD 89 CF 3A 2C
D8E0 : CD FE 01 C2 50 D9 32 03 CD CD 66 D0 CD 11 D1 DD
D8F0 : 7E 02 3C FE 04 FA F9 D8 AF DD 77 02 DD 66 04 DD
D900 : 6E 03 22 0A CE DD 66 06 DD 6E 05 22 0C CE CD 4C
D910 : D3 2A 1A CE DD 75 03 DD 74 04 3A 1E CE A7 CA 2A
D920 : D9 DD 7E 00 DD 86 07 DD 77 00 3A 1F CE A7 CA 38
D930 : D9 DD 7E 01 3C DD 77 01 CD 66 D0 CD AC D7 AF 32
D940 : 03 CD CD 11 D1 3A 07 CD A7 C2 50 D9 2A F4 CE 77
```

```
D950 : D1 1D C2 D6 D8 C9 00 00 2A F4 CE 22 9D CE CD 60
D960 : D5 2A 9D CE 22 F4 CE C9 3A 84 CE 3C 32 84 CE 47
D970 : 3A 81 CE B8 F2 7C D9 3E 01 32 85 CE CD C0 D5 C9
D980 : 1E 05 D5 7B 32 2B CD CD 6D CF 3A 2C CD FE 01 C2
D990 : 7A DB 3E 01 32 93 CE 32 03 CD CD D2 CF CD 11 D1
D9A0 : 3A 07 CD A7 C2 AD D9 CD C0 D5 C3 7A DB DD 7E 02
D9B0 : 3C DD 77 02 FE 08 C2 C3 D9 2A F4 CE 77 CD 68 D9
D9C0 : C3 7A DB 3A 9C CE DD 86 00 DD 77 00 CD D2 CF AF
D9D0 : 32 03 CD C3 70 DB D1 1D C2 82 D9 C9 1E 05 D5 7B
D9E0 : 32 2B CD CD A5 CF CD AD D0 3E 01 32 03 CD CD 11
D9F0 : D1 3A 2C CD FE 01 C2 59 DA FD 7E 02 00 E6 07 FD
DA00 : 77 02 FD 7E 03 32 0A CE FD 7E 04 32 0B CE FD 7E
DA10 : 05 32 0C CE FD 7E 06 32 0D CE CD 4C D3 2A 1A CE
DA20 : FD 75 03 FD 74 04 3A 1E CE A7 CA 36 DA FD 7E 07
DA30 : FD 86 00 FD 77 00 3A 1F CE A7 CA 44 DA FD 7E 01
DA40 : 3D FD 77 01 CD AD D0 AF 32 03 CD CD 11 D1 3A 07
DA50 : CD A7 C2 59 DA 2A F4 CE 77 D1 1D C2 DE D9 C9 00
DA60 : 3A 9A CE A7 C8 CD FC D0 D8 2A F4 CE CD 34 E4 3E
DA70 : 01 77 3A 8C CE FD 77 00 32 02 CE 3E 13 FD 77 01
DA80 : 32 03 CE 3E 27 32 00 CE 3E 03 32 01 CE CD F0 D2
DA90 : 3A 07 CE FD 77 07 3A 08 CE FD 77 05 3A 09 CE FD
DAA0 : 77 06 AF FD 77 03 FD 77 04 32 9A CE C9 FF FF FF
DAB0 : 3A 8F CE A7 C2 ED DA CD 50 D8 3E 01 32 03 CD CD
DAC0 : 11 D1 3A 99 CE 47 3A 8C CE 80 FE 45 FA D1 DA 3E
DAD0 : 45 FE 08 F2 D8 DA 3E 08 32 8C CE 3A 9A CE A7 C4
DAE0 : 60 DA CD 50 D8 AF 32 03 CD CD 11 D1 C9 CD A4 D8
DAF0 : AF 32 03 CD CD 11 D1 C9 AF 32 99 CE 32 9A CE 32
DB00 : 9C CE 3A 88 CE A7 C0 DB 00 47 E6 04 CA 16 DB AF
DB10 : 32 9B CE C3 25 DB 3A 9B CE A7 C2 40 E4 3E 05 32
DB20 : 9A CE 32 9B CE 00 78 E6 02 C2 31 DB 3E 02 32 9C
DB30 : CE 00 78 E6 08 C2 40 DB 3A 9C CE 3D 3D 32 9C CE
DB40 : DB 05 E6 04 C2 4E DB 3A 99 CE 3D CD 88 DB DB 02
DB50 : E6 08 C2 5C DB 3A 99 CE 3C CD 8D DB DB 09 E6 01
DB60 : CA DC DF C9 32 73 CE 3E 02 DD 77 04 C9 00 00 00
DB70 : CD 11 D1 3A 07 CD A7 CC C0 D5 D1 1D C2 82 D9 C9
DB80 : CD C0 D5 CD 34 CF C9 FF 3D 32 99 CE C9 3C 32 99
DB90 : CE C9 CD 32 09 CE 3A 04 CE A7 C0 AF 32 07 CE C9
DBA0 : 32 8C CE 3A 80 CE E6 0F FE 0C FA AF DB D6 0C CB
DBB0 : 27 4F 06 00 21 E8 CC 09 7E 32 81 CE 23 7E 32 82
DBC0 : CE C9 00 00 F5 E6 80 C2 CE DB F1 06 00 C9 06 FF
DBD0 : F1 C9 00 00 3A 0C CE CD C4 DB C9 00 3A 0D CE CD
DBE0 : C4 DB C9 00 3A 27 CD FE 12 D8 3E 11 32 27 CD C9
DBF0 : 00 3A 28 CD 32 1D CD FE 11 D8 E1 CD A4 CF C9 00
DC00 : 2A 1B CD 3A 1D CD FE 12 D8 CD 4A CF C9 00 FF FF
DC10 : 3A 7B CE E6 0F FE 0A F0 06 00 4F 21 F4 CB 09 4E
DC20 : 06 CB ED 43 08 CD 3A 79 CE 32 00 CD 3D 32 05 CD
DC30 : 3C 3C 32 12 CD 3A 7A CE 32 01 CD 3D 32 06 CD 3E
DC40 : 02 32 0B CD 32 0A CD 3E F8 32 04 CD AF 32 03 CD
DC50 : CD 11 D1 C9 3A 71 CE E6 0F 32 7B CE 3E 02 32 7A
DC60 : CE 3E 4C 32 79 CE CD 10 DC 3A 71 CE E6 F0 CA 79
DC70 : DC 3A 74 CE F6 02 32 74 CE 3A 74 CE E6 02 C8 3E
DC80 : 49 32 79 CE 3A 71 CE E6 F0 0F 0F 0F 0F 32 7B CE
DC90 : CD 10 DC 3A 72 CE E6 0F CA A4 DC 3A 74 CE F6 04
DCA0 : 32 74 CE 00 3A 74 CE E6 04 C8 3E 46 32 79 CE 3A
DCB0 : 72 CE E6 0F 32 7B CE CD 10 DC 3A 72 CE E6 F0 CA
DCC0 : CA DC 3A 74 CE F6 08 32 74 CE 3A 74 CE E6 08 C8
DCD0 : 3E 43 32 79 CE 3A 72 CE E6 F0 0F 0F 0F 0F 32 7B
DCE0 : CE CD 10 DC 3A 73 CE E6 0F CA F4 DC 3A 74 CE F6
DCF0 : 10 32 74 CE 3A 74 CE E6 10 C8 3E 40 32 79 CE 3A
DD00 : 73 CE E6 0F 32 7B CE CD 10 DC 3A 73 CE E6 F0 CA
DD10 : 1A DD 3A 74 CE F6 20 32 74 CE 3A 74 CE E6 20 C8
DD20 : 3E 3D 32 79 CE 3A 73 CE E6 F0 0F 0F 0F 0F 32 7B
DD30 : CE CD 10 DC C9 FF FF FF CD A1 DF A7 CA 49 DD AF
DD40 : 32 8F CE 3E 27 CD AB DD C9 3A 8C CE FE 27 CA 59
DD50 : DD F2 58 DD 3C C3 59 DD 3D F5 3E 01 32 03 CD CD
DD60 : 50 D8 CD 11 D1 CD D0 D4 F1 32 8C CE AF 32 03 CD
DD70 : CD 50 D8 CD 11 D1 3A 8C CE FE 27 C2 49 DD 3A 80
DD80 : CE 3C 32 80 CE 3A F6 CE 3C 27 32 F6 CE 3A 80 CE
DD90 : FE 0C FA 9A DD D6 0C C3 90 DD 4F 06 00 CD 88 DF
DDA0 : 09 7E 32 81 CE 23 7E 32 82 CE C9 32 8C CE 3A 91
DDB0 : CE 3D 32 91 CE C9 00 00 21 A2 C7 3A 8D CE 0E 00
DDC0 : BE D2 C9 DD 23 0C C3 C0 DD 06 00 21 9D C7 09 6E
DDD0 : 26 C7 22 08 CD 7E 32 0A CD 4E CB 39 3E 27 91 32
DDE0 : 05 CD 81 81 3C 32 12 CD 7E 32 0A CD 23 7E 32 0B
DDF0 : CD 3A 8D CE 4E CB 39 91 32 06 CD 3E F8 32 04 CD
DE00 : C9 00 00 00 3A 91 CE A7 FA DC DF CD 34 CF 00 CD
DE10 : C0 E0 3E 01 32 03 CD CD 11 D1 CD D0 D4 3A 8D CE
DE20 : 3D 32 8D CE CD B8 DD AF 32 03 CD CD 11 D1 3A 07
DE30 : CD A7 C2 0E DE CD 34 CF CD 4A CF 21 E4 CA 22 08
DE40 : CD 3E 0B 32 06 CD 3E 02 32 0B CD 3E 1C 32 05 CD
DE50 : 3E 31 32 12 CD 3E 14 32 0A CD 3E 58 32 04 CD AF
DE60 : 32 03 CD CD 11 D1 3A F6 CE E6 F0 0F 0F 0F 0F 32
DE70 : 7B CE 3E 0C 32 7A CE 3E 36 32 79 CE CD 10 DC 3A
DE80 : F6 CE E6 0F 32 7B CE 3E 39 32 79 CE CD 10 DC 00
DE90 : 00 00 3E FF CD 2A E4 00 CD 90 DF C9 CD 78 DF CD
DEA0 : 34 CF CD 4A CF CD E8 D7 3E 99 32 F6 CE AF 00 32
DEB0 : 7F CE CD 02 16 3A 76 EA 32 7D CE 3E FF 32 80 CE
DEC0 : 00 00 CD 38 DD CD 04 DE 00 00 00 AF 32 93 CE 32
DED0 : 71 CE 32 72 CE 32 73 CE 32 74 CE 32 84 CE 32 85
DEE0 : CE 32 86 CE 32 87 CE 32 88 CE 32 92 CE 32 93 CE
DEF0 : 3E 02 32 91 CE C9 CD 9C DE 00 3A 92 CE 47 3A 93
DF00 : CE B0 CA 09 DF 3E 00 D3 40 CD D0 D4 CD F8 DA CD
DF10 : B0 DA CD C5 DF 3A 92 CE A7 C2 21 DF D3 40 C3 25
DF20 : DF 3D 32 92 CE 3A 93 CE A7 CA 30 DF 3D 32 93 CE
```

```
DF30 : CD D4 D8 CD DC D9 CD 80 D9 3A 8F CE  7 CA 47 DF
DF40 : 3A 92 CE A7 CA 6C DF 21 2D CD 7E FE 02 C2 F9 DE
DF50 : 23 7E FE 02 C2 F9 DE 23 7E FE 02 C2 F9 DE 23 7E
DF60 : FE 02 C2 F9 DE 23 7E FE 02 C2 F9 DE CD 38 DD CD
DF70 : 04 DE C3 F9 DE 00 00 00 CD 00 CF 3E 05 32 91 CE
DF80 : C9 CD E8 D7 CD 34 CF C9 21 E8 CC CB 21 C9 FF FF
DF90 : CD 81 DF AF 32 8F CE 32 85 CE 32 87 CE 32 88 CE
DFA0 : C9 CD E5 DF 3A 8F CE C9 CD 7C D6 CD 54 DC C9 FF
DFB0 : 32 9C CE 01 14 00 21 A0 CE 11 2D CD ED B0 32 93
DFC0 : CE 32 92 CE C9 CD A8 DF 3E 03 32 79 CE 3E 02 32
DFD0 : 7A CE 3A 91 CE 32 7B CE CD 10 DC C9 AF D3 40 E1
DFE0 : C9 00 00 00 00 CD 34 CF AF D3 40 32 9C CE C9 00
DFF0 : 3A 00 CD 3D 3D 32 05 CD C6 05 32 12 CD 3A 01 CD
E000 : 3D 32 06 CD 21 05 03 22 0A CD C9 00 01 00 00 C5
E010 : 21 EE C7 09 6E 26 CB 22 08 CD 3E 0C 32 01 CD 21
E020 : F8 C7 09 7E 32 00 CD CD F0 DF 3E 58 32 04 CD CD
E030 : 4D E2 C1 0C 79 FE 08 C2 0F E0 C9 FF 3E 70 32 81
E040 : CE 3E 70 32 82 CE AF 32 84 CE 32 85 CE 32 8F CE
E050 : 00 CD DC E1 CD 7C D6 CD 80 D9 CD D4 D8 CD F0 D3
E060 : 3A 97 CE 87 32 9C CE CD F0 D3 3A 97 CE 87 32 99
E070 : CE CD F0 D3 3A 94 CE E6 0F CA 7F E0 32 9A CE CD
E080 : DC D9 CD B0 DA AF 32 9A CE 3A 8F CE A7 CA 50 E0
E090 : CD 34 CF AF 32 99 CE 32 9C CE 3E 30 32 65 CE 00
E0A0 : F5 CD DC E1 CD 7C D6 CD 80 D9 CD DC D9 CD B0 DA
E0B0 : F1 3D C2 A0 E0 CD 34 CF C9 FF FF FF FF FF FF FF
E0C0 : CD B8 DD CD D0 D4 C9 00 01 00 00 00 C5 21 FE CB
E0D0 : 09 6E 26 CB 22 08 CD 3E 04 32 01 CD 21 06 CC 09
E0E0 : 7E 32 00 CD CD F0 DF AF 32 03 CD 79 CB 27 E6 07
E0F0 : 4F 21 E0 CC 09 7E 32 04 CD CD 11 D1 3E 20 F5 CD
E100 : D0 D4 F1 3D C2 FE E0 C1 0C 79 FE 08 C2 CC E0 C9
E110 : 2A 08 CD 7E 32 0A CD CB 3F 57 3A 00 CD 82 32 05
E120 : CD 57 3A 0A CD 82 3C 32 12 CD 3E 0B 32 06 CD 3E
E130 : 05 32 0B CD 3E 58 32 04 CD AF 32 03 CD CD 11 D1
E140 : CD D0 D4 C9 01 00 00 00 C5 21 B8 CC 09 6E 26 CC
E150 : 22 08 CD 3E 0A 32 01 CD 21 BD CC 09 7E 32 00 CD
E160 : CD 10 E1 C1 0C 79 FE 05 C2 48 E1 C9 CD C8 E0 CD
E170 : 44 E1 3E FF F5 CD D0 D4 F1 3D C2 74 E1 C9 00 00
E180 : 3E 05 32 05 CD 3E 01 32 06 CD 3E 20 32 12 CD 21
E190 : 1B 02 22 0A CD 3E D8 32 04 CD 21 AC CA 22 08 CD
E1A0 : AF 32 03 CD CD C8 E1 2A 75 CE 22 71 CE 2A 77 CE
E1B0 : 22 73 CE CD 54 DC 21 00 00 22 71 CE 22 73 CE C9
E1C0 : CD 80 E1 CD 3C E0 C9 00 3E 27 32 00 CD 3E 02 32
E1D0 : 01 CD AF 32 03 CD CD 11 D1 C9 00 00 CD D0 D4 CD
E1E0 : 80 E1 C9 00 CD 00 CF CD 34 CF CD 4A CF CD E8 D7
E1F0 : CD 6C E1 CD 34 CF CD 3C E0 CD E8 D7 CD 34 CF C9
E200 : CD E4 E1 CD F6 DE CD 35 CF CD 55 E2 CD 54 DC A7
E210 : 3A 77 CE 47 3A 73 CE B8 CA 22 E2 DA 38 E2 00 C3
E220 : 00 E4 3A 76 CE 47 C3 E2 E3 00 00 22 75 CE 2A 73
E230 : CE 22 77 CE CD 80 E1 00 CD D0 D4 DB 09 FE FF CA
E240 : 38 E2 E6 01 C2 00 E2 CD 60 CF C3 66 5C AF 32 03
E250 : CD CD 11 D1 C9 CD 4A CF CD 0C E0 C9 CD B2 BA B3
E260 : B2 C4 DE B3 20 20 20 20 3C 2D 28 5A 29 20 20 20
E270 : 28 43 29 2D 3E BE DD B6 B2 B2 C4 DE B3 20 20 20
E280 : 20 3C 2D 28 31 29 20 20 20 28 33 29 2D 3E BA B3
E290 : BC B7 DE AE D7 B2 20 CA AF BC AC 20 20 20 28 32
E2A0 : 29 B8 D8 DD BA DE DD 20 C9 20 C3 DD BD B3 BC DB
E2B0 : 20 20 20 A5 A5 A5 20 31 30 C3 DD D0 C4 DE D8 20
E2C0 : A5 A5 A5 20 32 30 C3 DD D0 BD DE B2 DB A5 A5 A5
E2D0 : 20 33 30 C3 DD B7 B2 DB 20 20 A5 A5 A5 20 34 30
E2E0 : C3 DD B2 CF 20 CA 20 D6 D9 20 C3 DE BD B6 3F 28
E2F0 : 59 2F 4E 29 3A 01 CD 3C 32 63 EA 3A 00 CD 3C 32
E300 : 64 EA 3A 02 CD 2A 08 CD 00 F5 E5 7E CD 57 02 E1
E310 : 23 F1 3D C2 08 E3 C9 00 21 18 01 22 5D EA 01 19
E320 : 24 CD 3A 09 3E 0C CD 57 02 3E 05 32 00 CD 3E 05
E330 : 32 01 CD 3E 19 32 02 CD 21 5C E2 22 08 CD CD F4
E340 : E2 3E 07 32 01 CD 21 75 E2 22 08 CD CD F4 E2 3E
E350 : 09 32 01 CD 3E 13 32 02 CD 21 8E E2 22 08 CD CD
E360 : F4 E2 3E 0C 32 01 CD 3E 0D 32 02 CD 21 A1 E2 22
E370 : 08 CD CD F4 E2 3E 0E 32 01 CD 3E 07 32 00 CD 21
E380 : AE E2 22 08 CD CD F4 E2 3E 10 32 01 CD 21 BB E2
E390 : 22 08 CD CD F4 E2 3E 12 32 01 CD 21 C8 E2 22 08
E3A0 : CD CD F4 E2 3E 14 32 01 CD 21 D5 E2 22 08 CD CD
E3B0 : F4 E2 3E 16 32 01 CD 3E 12 32 02 CD 21 E2 E2 22
E3C0 : 08 CD CD F4 E2 CD 75 0F 3A 69 EA CD C1 5F FE 59
E3D0 : C2 DA E3 AF 32 06 DF C3 00 E2 3E 20 32 06 DF C3
E3E0 : 00 E2 3A 72 CE B8 CA EF E3 DA 38 E2 C3 00 E4 3A
E3F0 : 75 CE 47 3A 71 CE B8 CA 00 E4 DA 38 E2 00 00 00
E400 : 2A 71 CE C3 29 E2 CD F0 D3 3A 94 CE E6 01 2A 8C
E410 : CE C8 DD 46 01 3E 18 90 47 3A 99 CE A7 3A 8C CE
E420 : C8 F2 27 E4 90 6F C9 80 6F C9 F5 CD D0 D4 F1 3D
E430 : C2 2A E4 C9 3A 7E CE E6 07 FD 77 02 C9 FF FF FF
E440 : 3D 32 9B CE C3 25 DB 00 E5 01 90 F4 A7 ED 42 E1
E450 : C8 A7 CA 63 D4 C3 66 D4 ED 4B 10 CD 03 21 A0 CE
E460 : 1B 03 ED B0 C3 73 D1
```



◉チェック・サム プログラムの使い方

プログラムをＲＵＮさせてから，スタート番地，エンド番地を16進数で入力します．バイト数は64バイトと指定してください．

チェック・サムはモニタとプリンタへ同時に出力されるので，プリンタのない方は行番号240を削除してください．

また，Ｃ001番地より以前にあるプログラムのチェックを取る場合は，行番号100の＆HC5FFを適当に変えてください．

チェック・サム　プログラム・リスト

```
100 CLEAR300,&HC5FF
110 WIDTH40,25:CONSOLE0,25,0,0
120 PRINTCHR$(12):PRINT"PC-8001 check "
130 INPUT"start=";A$:S=VAL("&h"+A$):IF S<0THEN S=S+65536!
140 INPUT"end=";B$:E=VAL("&h"+B$):IF E<0THEN E=E+65536!
150 INPUT"byte =";T
160 X=0:R=S+T:IF R>=E THEN R=E+1
170 FOR I=S TO R-1
180 Y=PEEK(I)
190 X=X+Y:NEXT
200 K=S:GOSUB260:S$=K$
210 S=S+T:K=R-1:GOSUB260:E$=K$
220 K=X:GOSUB260:X$=K$
230 PRINTS$" - ";E$" = ";X$
240 LPRINTS$;" - ";E$;" = ";X$
250 IF S>=E THEN END ELSE GOTO 160
260 U=INT(K/4096):K=K-4096*U:K$=HEX$(U)
270 U=INT(K/256):K=K-256*U:K$=K$+HEX$(U)
280 U=INT(K/16):K=K-16*U:K$=K$+HEX$(U)+HEX$(K)
290 RETURN
```



マシン語プログラムのチェック・サム

C600 - C63F = 12B1	CC80 - CCBF = 1ACF	D300 - D33F = 1E42	D980 - D9BF = 231B	E000 - E03F = 1A2F
C640 - C67F = 1179	CCC0 - CCFF = 1BE2	D340 - D37F = 2138	D9C0 - D9FF = 2211	E040 - E07F = 27D6
C680 - C6BF = 15B6	CD00 - CD3F = 0EFB	D380 - D3BF = 1B55	DA00 - DA3F = 1DDF	E080 - E0BF = 2BD3
C6C0 - C6FF = 12F6	CD40 - CD7F = 0E8B	D3C0 - D3FF = 1D4E	DA40 - DA7F = 2271	E0C0 - E0FF = 1C77
C700 - C73F = 095C	CD80 - CDBF = 1007	D400 - D43F = 1F8A	DA80 - DABF = 1F9F	E100 - E13F = 1C34
C740 - C77F = 0CE6	CDC0 - CDFF = 12DC	D440 - D47F = 1D19	DAC0 - DAFF = 23AE	E140 - E17F = 2050
C780 - C7BF = 0E87	CE00 - CE3F = 0FA7	D480 - D4BF = 20D3	DB00 - DB3F = 1FE4	E180 - E1BF = 194A
C7C0 - C7FF = 10AE	CE40 - CE7F = 105F	D4C0 - D4FF = 21B1	DB40 - DB7F = 2216	E1C0 - E1FF = 23E2
C800 - C83F = 0F67	CE80 - CEBF = 0982	D500 - D53F = 1EDF	DB80 - DBBF = 20F1	E200 - E23F = 25DC
C840 - C87F = 0A3E	CEC0 - CEFF = 00FA	D540 - D57F = 2186	DBC0 - DBFF = 20DF	E240 - E27F = 1F07
C880 - C8BF = 0FED	CF00 - CF3F = 1A3E	D580 - D5BF = 210B	DC00 - DC3F = 1BE9	E280 - E2BF = 1E9F
C8C0 - C8FF = 197A	CF40 - CF7F = 1D83	D5C0 - D5FF = 1F7D	DC40 - DC7F = 1E8C	E2C0 - E2FF = 1FBC
C900 - C93F = 0CBA	CF80 - CFBF = 1D77	D600 - D63F = 21C1	DC80 - DCBF = 200C	E300 - E33F = 17D1
C940 - C97F = 0DA2	CFC0 - CFFF = 1BA7	D640 - D67F = 209A	DCC0 - DCFF = 1FE1	E340 - E37F = 1981
C980 - C9BF = 072C	D000 - D03F = 1931	D680 - D6BF = 2163	DD00 - DD3F = 23DA	E380 - E3BF = 1DF5
C9C0 - C9FF = 1420	D040 - D07F = 19D3	D6C0 - D6FF = 250A	DD40 - DD7F = 23DC	E3C0 - E3FF = 229E
CA00 - CA3F = 0C04	D080 - D0BF = 1DAD	D700 - D73F = 2324	DD80 - DDBF = 1F5C	E400 - E43F = 25E4
CA40 - CA7F = 0E52	D0C0 - D0FF = 1C3E	D740 - D77F = 1C2E	DDC0 - DDFF = 1AAC	E440 - E466 = 1592
CA80 - CABF = 13AB	D100 - D13F = 20D6	D780 - D7BF = 20B5	DE00 - DE3F = 218F	
CAC0 - CAFF = 19D1	D140 - D17F = 1C06	D7C0 - D7FF = 1EC8	DE40 - DE7F = 17BC	
CB00 - CB3F = 153B	D180 - D1BF = 1D38	D800 - D83F = 2474	DE80 - DEBF = 2214	
CB40 - CB7F = 1D4A	D1C0 - D1FF = 1A92	D840 - D87F = 1C20	DEC0 - DEFF = 1DEC	
CB80 - CBBF = 211B	D200 - D23F = 1C54	D880 - D8BF = 179E	DF00 - DF3F = 283F	
CBC0 - CBFF = 1AEF	D240 - D27F = 1C44	D8C0 - D8FF = 1E4F	DF40 - DF7F = 2401	
CC00 - CC3F = 143C	D280 - D2BF = 1CC2	D900 - D93F = 1E25	DF80 - DFBF = 26C0	
CC40 - CC7F = 2176	D2C0 - D2FF = 20DF	D940 - D97F = 2367	DFC0 - DFFF = 1D4A	



インベーダーブームが去り，ヘッド・オンは何かパッとしなくて，約半年前（？）から出現したのがギャラクシアンでした．一度途中まで作ってみたときはメモリ・オーバーになってしまい，あきらめていました．しかし，誰も本に発表してくれそうもなく，奮起して今度は省エネルギならぬ省メモリに徹し，さらに最近の知識（のつもり）を投入して16K RAMでも走るギャラクシアンを完成させました．

1．本物と比べて

最近のギャラクシアンは，パートIIとかパートIVとか，かなり進んでいるようですが，何しろ原型しかやったことがないので，それに準じたものにしました．

本物に比べておつるところは，

❶あのすばらしい音が出ない．
❷ドットが荒いのでちゃんとした形にならない．
❸背景の星の流れがない．
❹ライン・カラーになっていて，各ギャラクシアンごとに色がつかない．
❺カーブでギャラクシアンが傾かない．

など，いろいろあります．ミサイルが矢ブスマのごとく降ってくる点など，本物以上にむずかしいところもあります．面白さのエッセンスだけは取り入れてあるつもりです．

2．プログラムの入力について

BASICのプログラムにはチェック・サムが付属しています．BASIC，マシン語とも入力したらテープに入れておきましょう．マシン語は間違えるとたいてい暴走します．

RUNすると，

```
GALAXIAN CHECK SUM START=D400 END=E495
```

と表示した後，次々と64バイトごとのチェック・サムの値が出てきます．適当なところで［ESC］キーで画面を止めて，本の値と比べてください．もし違っているときはそこの番地の範囲についてチェックしてください．

プリンタのある人は，行番号5090のPRINTをLPRINTに直せば，プリント出力できます．

チェック・サムがすべてOKなら，恐らくマシン語は大丈夫と思われます．BASICの行番号10と5000～5140を消して，もう一度テープにとっておきましょう．

3．ゲームの解説

ビーム砲はテンキーの［4］で左へ，［6］で右へ移動します．ビームの発射は［スペース］キーです．

RUNするとまず解説が出てきて，［RETURN］キーを押すとゲームが始まります．画面右側には，SCORE，Hi-SCORE，ビーム砲の数が表示されます．MYSTERY POINTというのは攻撃中の母船に命中したとき表示されます．SCENEは局面数のことです．

コンボイが移動を始め，しばらくするとビーム砲が描かれて開始です．この時間はビーム砲がやられたとき空中にいるギャラクシアンがコンボイへ戻るためで，この時間なしにすぐにビーム砲を描くと，訳のわからないうちに再びやられるはめになります．

最初のうちは1台のギャラクシアンが攻撃を仕掛けてきます．下半分にかかるとミサイルを発射してくるので，避けながら撃破してください．ギャラクシアンの数が減るにつれて攻撃してくる数が増えてきます．母船と護衛はたいていの場合，3台で1グループになって攻撃してきます．

ギャラクシアンが全部やられたとき，あるいはコンボイ型が1台残っているときに局面は次に移ります．

5,000点を超すとビーム砲は1台追加されます．このときの追加音が，ビーム砲をやられたときの音に似ていますので，間違えないようにしてください．

25面消し終わるとギャラクシアンは全滅でGIVE UPしますが，ここまで行く人は……まずいないでしょう．

4．得点

得点は**表1**に示します．攻撃型はコンボイ型の倍になっています．MYSTERY POINTは母船のみに命中したとき200点，護衛に命中した後，母船に命中したとき400点，護衛2機に命中してから母船に命中したとき800点で，このときの得点は画面右側にしばらく表示されます．

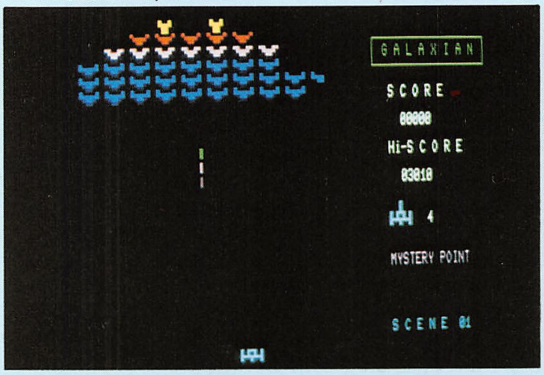
RUN直後に表示されるゲームの説明



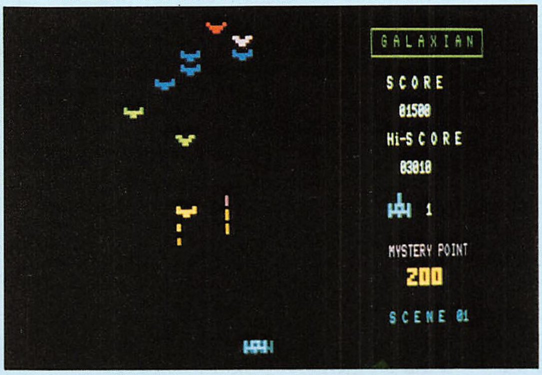
こうなると逃げ回るのも大変！



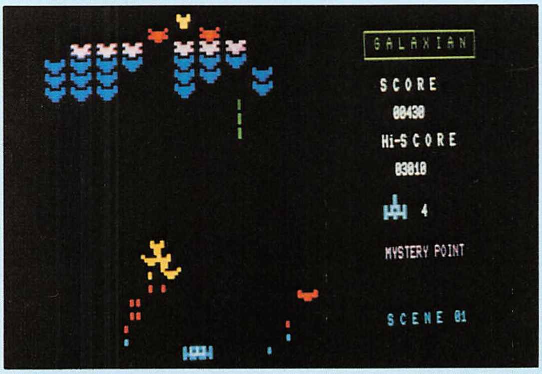
ゲームを開始して，しばらくはギャラクシアンの攻撃がないが…



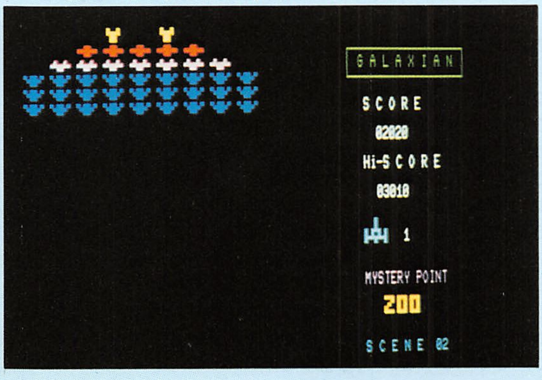
母船をうまく攻撃してミステリーポイントがでる



時々，母船を含む3機が編隊を組んで攻撃してくる．



第1面を無事きりぬけ，第2面が始まったところ



5. プログラムの解説

❶マシン語

動きを必要とするところはすべてマシン語で組んであり，BASICへは終了時に戻るだけです．

図1にジェネラル・フローチャートを示します．16進の数値はサブルーチンの番地です．(1)～(7)のサブルーチンについては，それぞれのフローチャートを図2～図8に示します．これらのフローチャートを見れば，だいたいの動き

表1　得点

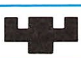
ギャラクシアンのタイプ	コンボイ型	攻撃型
	30	60
	40	80
 護衛	50	100
 母船	60	MYSTERY

表2　マシン語サブルーチン

番　　地	内　　　容
D400～D462	ギャラクシアンを描くサブルーチン
D463～D4B9	初期画面
D4BA～D4EC	10進出力サブルーチン
D4ED～D4FF	0～FFの乱数サブルーチン
D500～D50C	タイマ
D50D～D52F	クロック更新.［スペース］キーを戻したか.
D530～D5A8	ギャラクシアンの踊り
D5A9～D634	ギャラクシアンの左右移動
D635～D694	ビーム砲移動
D695～D6DB	ビーム発射
D6DC～D763	ビーム上昇, 命中. V-RAM内かコンボイ型.
D764～D7EB	〃　　　攻撃型
D7EC～D845	〃　　　母船
D846～D88C	コンボイ型のワークを消す.
D88D～D8C3	ギャラクシアン爆発マーク, 得点計算
D8C4～D909	ビーム砲追加したか?
D90A～D93F	攻撃型のワークを消す.
D940～D978	母船得点表示
D979～D9CF	母船得点の数字を描くサブルーチン
D9D0～D9DF	ビームを砲台につけるサブルーチン
D9E0～DA09	番地計算サブルーチン
DA0A～DA2B	ギャラクシアンの数から, 攻撃するものの数を決める.
DA2C～DACF	攻撃するギャラクシアンを決めて, データを移す.
DAD0～DAE7	ギャラクシアンを描くサブルーチン
DB56～DB7F	ギャラクシアン飛行のメイン・ルーチン
DB80～DB9F	データのエクスチェンジ
DBA0～DC36	ギャラクシアン飛行のサブルーチン
DC37～DC85	飛行コースのポインタ変更(1)
DC86～DCAB	ミサイル発生
DCAC～DD06	ビーム砲爆発マーク, ビーム砲を1台減らす.
DD07～DD5C	〃　　〃　のサブルーチン 音を出すサブルーチン
DD5D～DDCF	飛行コースのポインタ変更(2)
DDD0～DDEF	0～7の乱数を得るサブルーチン
DDF0～DE3F	コンボイ型へ戻る.
DE40～DE8C	攻撃型母船のデータ移動
DE8D～DEEC	〃　護衛の　〃
DEED～DF03	護衛のみの飛行
DF04～DF4B	攻撃型母船と護衛を決める.
DF4C～DF59	母船, 護衛のワークを消す.
DF5A～DF6C	飛行のコースを決める.
DF6D～DF7F	データのエクスチェンジ
DF80～DF9F	母船の飛行のメイン・ルーチン
DFA0～E049	母船と護衛の飛行サブルーチン
E04A～E068	コンボイ中の母船を描く.
E069～E07F	飛行コースのポインタ変更
E080～E0F9	ミサイル降下
E0FA～E12F	ミステリーポイント表示を消す.
E130～E165	ハイスコア・サブルーチン
E166～E17F	画面左半分のクリア
E180～E1A9	ギャラクシアン＝0かのチェック
E1AA～E1BF	RUN時初期設定
E1C0～E259	第1, 第2初期設定
E35A～E379	ビーム砲爆発時の初期設定
E37A～E415	メイン・ルーチン
E416～E41F	画面右のビーム砲を描く
E420～E495	カラー化

表3　ワーク・エリア

番　　地	内　　　容
E866	ビーム砲表示までの待機カウンタ
E867, E868	母船(1)飛行コース・ポインタ
E869, E86A	母船(2)　〃　　〃
E86B	母船 (攻撃型) の存在
E86C, E86D	〃　, 〃　のアドレス
E86E	〃　〃　　Y 値
E86F	〃　〃　　X 値
E870	〃　〃　　タイプ
E871, E872	〃　〃　　コンボイ中での位置
E873, E874	〃　〃　飛行コースポインタアドレス
E875～E87E	護衛(1)――母船と同様の配列
E87F～E888	〃 (2)――　〃　〃
E889, E88A	0～FFFFの乱数値
E88B	局面数
E88C	ミサイル(1)の存在
E88D, E88E	〃　(1)のアドレス
E88F～E8C7	ミサイル(2)～ミサイル(20)で, ミサイル(1)と同様
E8C8	ビーム砲を追加したが
E8C9	攻撃型ギャラクシアンの数
E8CA	攻撃型ギャラクシアン(1)の存在
E8CB, E8CC	〃　のアドレス
E8CD	〃　のY値
E8CE	〃　のX値
E8CF	〃　のタイプ
E8D0, E8D1	〃　のコンボイ中の位置
E8D2, E8D3	〃　の飛行コース・ポインタ・アドレス
E8D4～E8F1	攻撃型ギャラクシアン(2)～(4)で, (1)と同様
E8F2	ミステリーポイント表示カウンタ
E8F3	ミステリーポイント表示しているか?
E8F4, E8F5	HI-SCORE
E8F6, E8F7	SCORE
E8F8	コンボイ型のワーク内にあったかのフラグ
E8F9	護衛を何台やられたか?
E8FA	ビーム砲の数
E8FB, E8FC	ビーム・アドレス
E8FD	ビーム発射しているか?
E8FE	コンボイ型エイリアン(1)の存在
E8FF, E900	〃　　〃　(1)アドレス
E901	〃　　〃　(1)Y 値
E902	〃　　〃　(1)X 値
E903	〃　　〃　(1)タイプ
E904～E9F3	コンボイ型エイリアン(2)～(41)で, (1)と同様
E9F4	移動方向値一時スタック
E9F5, E9F6	ビーム砲アドレス
E9F7	ビーム砲やられたか?
E9F8	BASICリターン要求
E9F9	0～FFの乱数値
E9FA	ギャラクシアンの数
E9FB	〃　　の踊り
E9FC	〃　　移動方向
E9FD	8進クロック
E9FE	メイン・クロック
E9FF	［スペース］キーを戻したか?

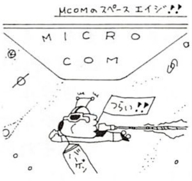


図１　ジェネラル・フローチャート

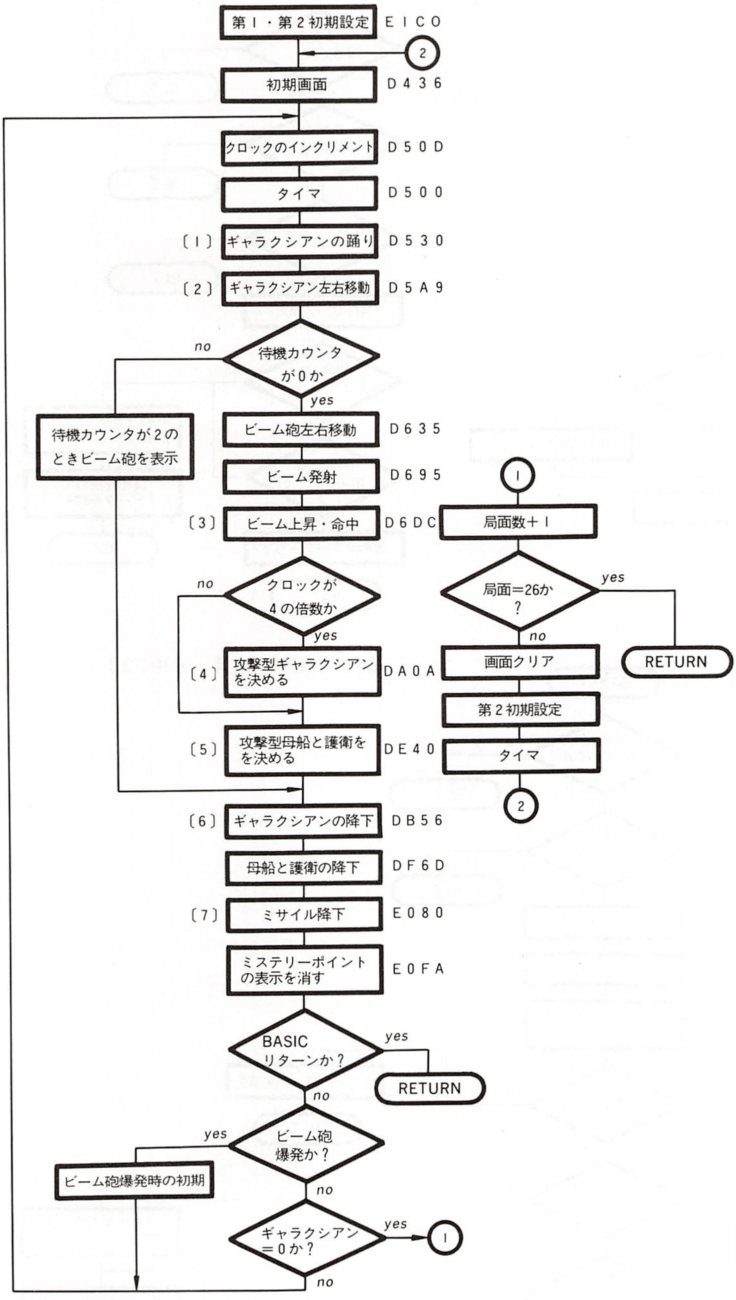


図５　攻撃型ギャラクシアン〔４〕

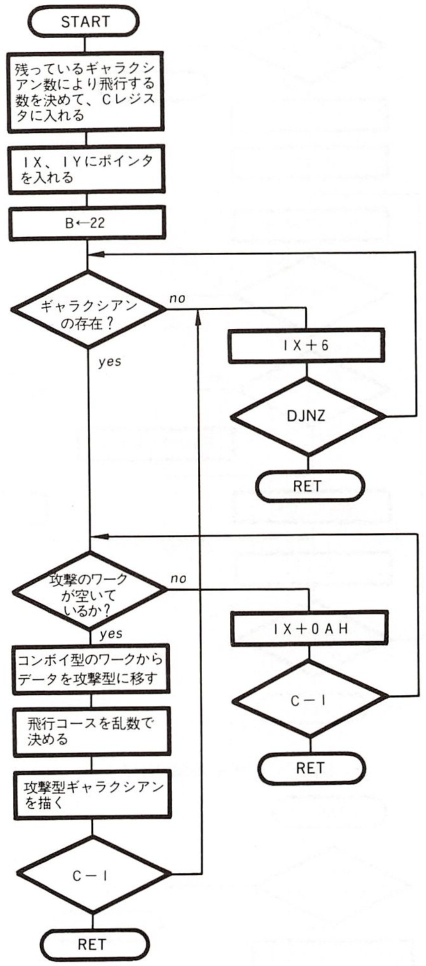


図２　ギャラクシアンの踊り〔１〕

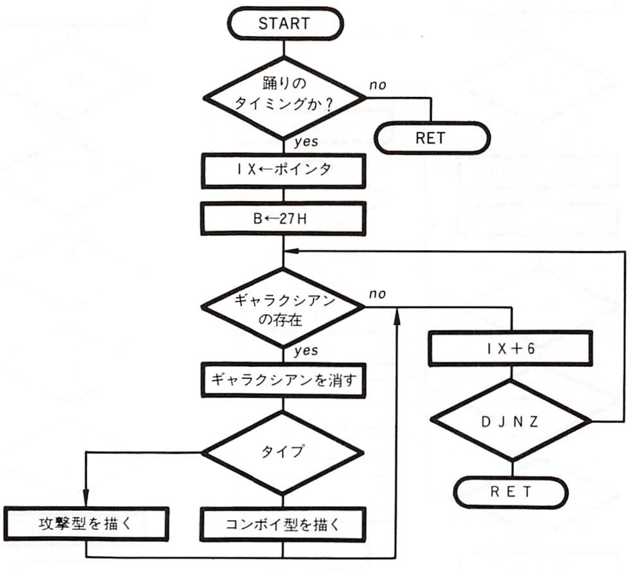


図3　ギャラクシアン左右移動〔2〕

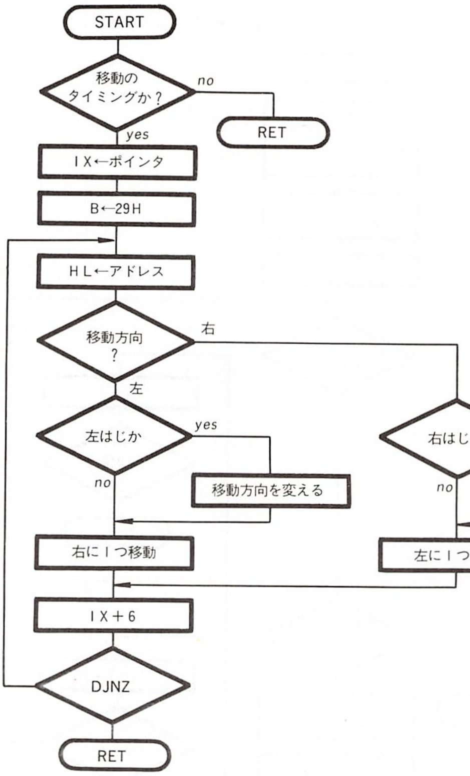


図6　攻撃型母船と護衛を決める〔5〕

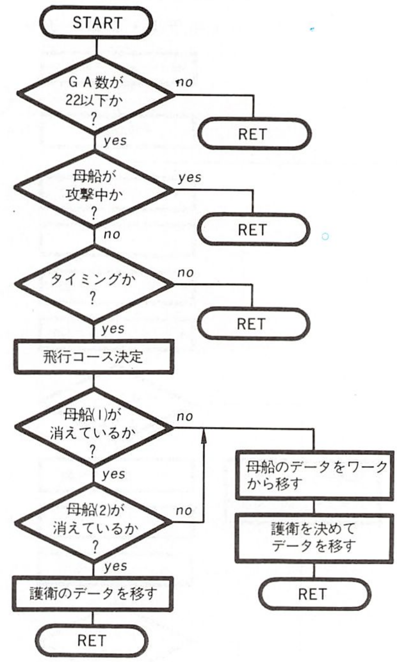


図4　ビーム上昇・命中〔3〕

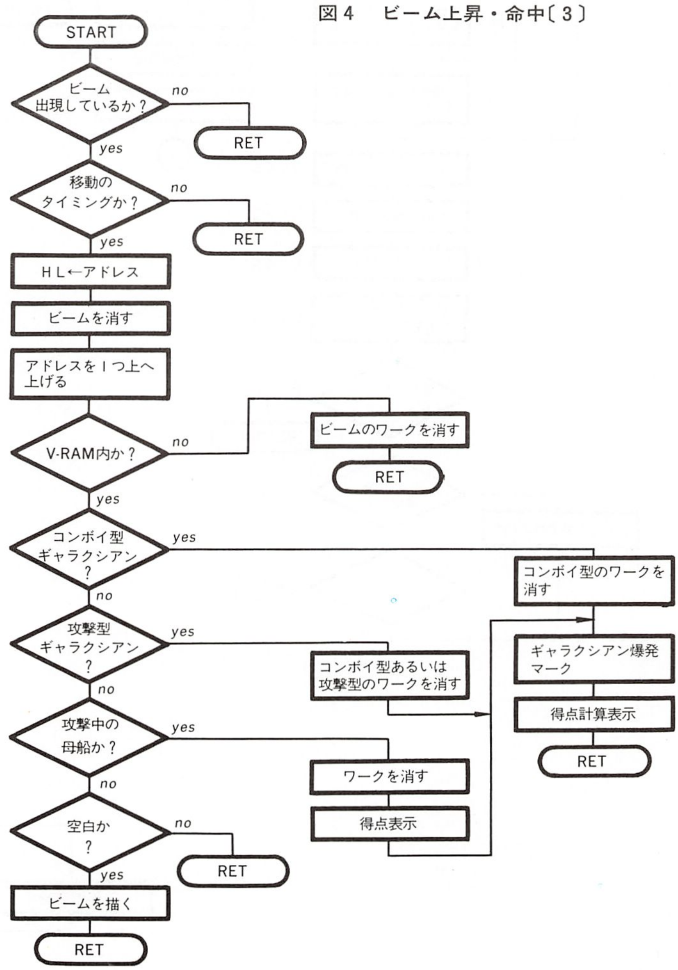


図8　ミサイル降下〔7〕

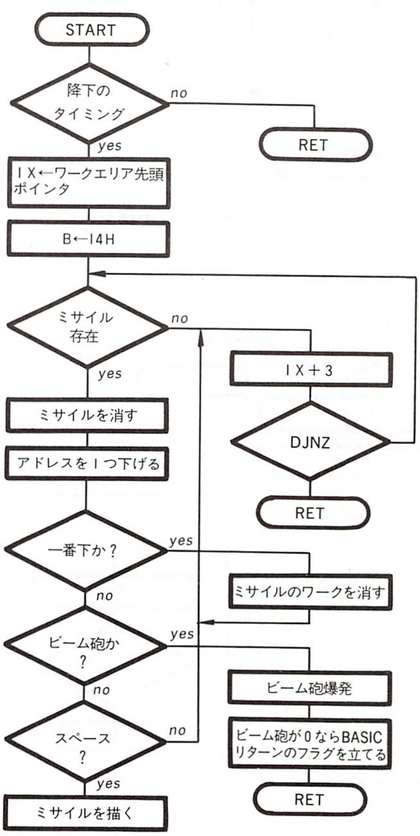
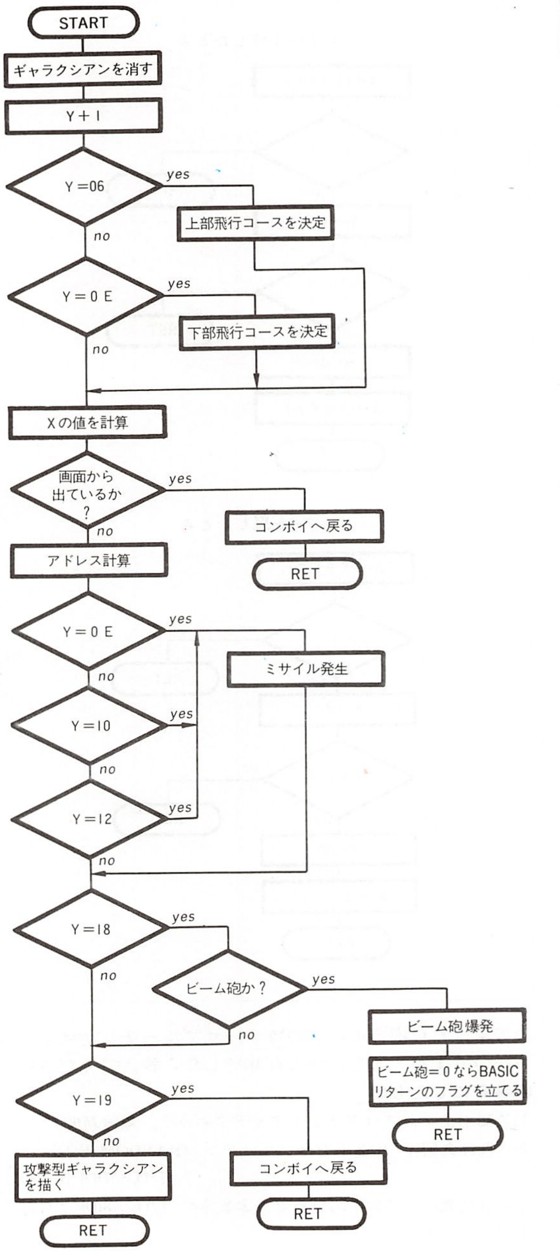


図7　ギャラクシアンの降下〔6〕



はわかると思います．

さらに詳しいサブルーチンの役割については**表2**を見てください．ワーク・エリアを**表3**に，データ・エリアを**表4**に示します．

マシン語の解説などは細かく書いていたらキリがないのでこの辺にしておきますが，フローチャートを見てそれから内容をみれば，理解は容易と思います．ただし，プログラムは必ずしもフローチャート通りとは限りませんが……．

❷BASIC

BASICのプログラムは，説明文，初期の表示，終了の表示くらいなので，簡単にわかると思います．プログラムの内容については**表5**に示します．

表4　データ・エリア

E25A～E359	初期設定のコンボイ型のデータ
DAE8～DB55	ギャラクシアン飛行コースのデータ

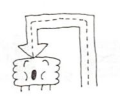
表5　BASICプログラム

行番号	内容
140～160	初期設定
170	説明文へ
180～200	画面のカラー化
210～280	画面右側の表示
290	マシン語　メイン・ルーチンへ
300	マシン語　ハイスコア・ルーチンへ
310	GAME OVERか，GIVE UPかの分岐
320～360	GAME OVERの表示
370～450	説明文
460～530	GIVE UPの表示
5000～5140	チェック・サム

表6　変更点

変更する点	番地	現在の値(16進数)
(1)全体のスピード	E384	05
※(2)ギャラクシアンの踊り速度	D534	07
※(3)ギャラクシアン左右移動速度	D5AD	1F
※(4)ビーム砲移動速度	D639	03
※(5)ビームの速度	D6E6	01
(6)ギャラクシアン飛行速度	D516	08
※(7)ミサイル速度	E084	03
(8)終了局面数	E3F0	1A
(9)ビーム砲の数	E1D1	04
(10)ミサイルの数	E08B	14

❻. 変更点



うまくなり過ぎてつまらないという人や（まずいないと思いますが），むずかしすぎてつまらないという人のために変更点を**表6**に示します．番号の頭に＊のついたものは変更する数値が2の倍数（00，01，03，07，0F，1F，3F，……）になるようにしてください．

（5）のビームの速度は，00にすると速くなりますが少し速すぎます．

（6）のギャラクシアン飛行速度は，04以下にはしないでください．また，あまり遅くすると，❸で説明したように待機時間のうちにコンボイへ戻れず，ビーム砲出現とかちあうことになります．

（7）のミサイル速度は，**（6）**との兼ね合いが大切で，ミサイルがギャラクシアンと同じ速度や逆に遅くては，変なものです．

（8）の終了局面数は，得点が最高65,535点までで，それ以上は0点に戻るため，それ以内に終了ということで25面に設定しました．もっとも25面まで行ったことがないので，実際に何点になるかは不明です．

（9）のビーム砲の数は，10台以上になると正しく表示されません．

（10）のミサイル数は現在20個ありますが，画面上にこんなにたくさん出現することはなく，10個くらいでよさそうです．プログラムを組む時点では見当がつかないので多めにしておいたのですが……．初心者のうちは，この数を少なくしておくと楽になると思います．

変更は，全体の調和を乱さないようにすることが重要で，調和のとれていないゲームは興味も半減します．

7.グリーン・モニタの人へ

本プログラムはカラー用にしてあるため，グリーン・モニタでは薄くて見えない部分もあると思います．そこで変更点ですが，まずマシン語ではE455番地(現在00)にC9を入れます．BASICでは，COLOR文の数値をすべて7にします．行番号260，270のLINE文中の数値も変えるのを忘れないようにしてください．

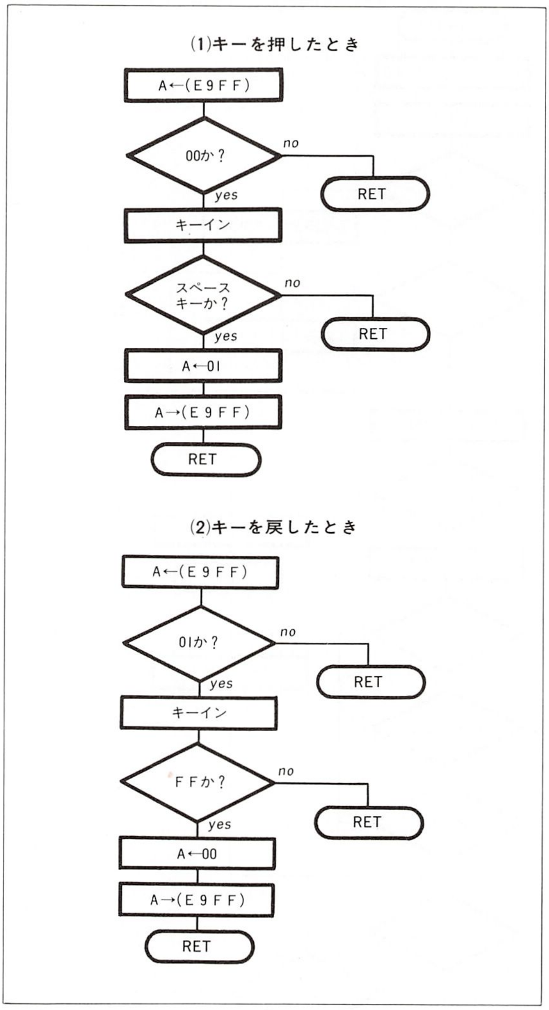
8.ワンポイント・アドバイス

—連続してキーの値を読まなくする方法—

ゲームでよく使うミサイルやビームの発射などのときマシン語でキーの読み込みをすると，押している間連続して読めるので，押しているだけで連続してビームが出てしまいます．これでは興味半減で，やはり本物のように1回ごとにキーを押すようにしたいものです（もっとも，キーは早く痛む！）．

さてその方法ですが，本ゲーム中でも使用しているもので，押すキーは［スペース］キー，ワーク・エリアではE9FFに初期値として00が入っているものとします．キーを離してないときはここに01が入ることになります．ここでは，フローチャートの形で図9に示しておきます．これら2つのサブルーチンは，メインのループの中のどこにあってもかまいません．本プログラムでは（1）はD695～D6AA，（2）はクロック更新サブルーチン内で，D51D～D52Cにあるので，実際のマシン語はそちらをみてください．

図9　連続してキーの値を読まなくする方法



9.おわりに

やはりマシン語のプログラムは時間のかかるもので，以前に作りかけたという基礎があるにもかかわらず，かなりの日数がかかりました．BASICのようにスラスラ組めるようになるのはいつのことやら．早くアセンブラが欲しい！

高得点のコツは……，ただひたすらやるのみという感じです．とにかく高得点のギャラクシアンを狙うなんてことより，ひたすらミサイルにあたらないよう逃げまわることが高得点への道かもしれません．筆者の今までのHI-SCOREは11,140点でした．

カラーはラインごとのカラーで，キレイではありますが，やはりギャラクシアン1機ごとに固有の色をつけたく，それにはアトリビュート・エリアをいじるという難関があるのです．PC内のモニタのカラーサブルーチンが使えるといいのですが，どなたかご存知でしたら教えてください．

●参考にしたもの

1)Z80マイコンプログラミング・テクニック，電波新聞社
2)八丸鉢真留：〝10進出力サブルーチン（RANDOM BOX)〟 I/O，'79年10月号
3)阿蘇坊舞子：〝舞子のプログラム教室〟 I/O，'80年7月号
4)ゲーム：ギャラクシアン

BASICプログラム

```
10 GOTO 5000
100 ' ┌──────────────────────────────────┐
110 ' |   GALAXIAN        VER 1.4          |
120 ' |                        1980年7月16日 |
130 ' └──────────────────────────────────┘
140 CLEAR300,&HD3FF
150 CONSOLE0,25,0,1:WIDTH80,25:COLOR7,0,1:PRINTCHR$(12)
160 DEFUSR2=&HE1AA:AA=USR2(0)
170 GOSUB370
180 COLOR7,0,1:PRINTCHR$(12)
190 LINE(2,0)-(2,24)," ",B:LINE(50,0)-(50,24)," ",B:LINE(56,0)-(56,24)," ",B
200 DEFUSR5=&HE42B:AA=USR5(0)
210 COLOR4:LOCATE57,1:PRINT"┌─────────────┐"
220 LOCATE57,2:PRINT"| G A L A X I A N |"
230 LOCATE57,3:PRINT"└─────────────┘"
240 COLOR7:LOCATE60,5:PRINT"S C O R E":LOCATE62,7:PRINT"00000"
250 LOCATE60,9:PRINT"Hi-S C O R E":LOCATE62,11:PRINT"     ":DEFUSR3=&HE157:AA=US
R3(0)
```

チェック・サム

```
D400 - D43F = 0ECA
D440 - D47F = 158D
D480 - D4BF = 1B88
D4C0 - D4FF = 19FE
D500 - D53F = 2037
D540 - D57F = 168C
D580 - D5BF = 197D
D5C0 - D5FF = 16A4
D600 - D63F = 17FD
D640 - D67F = 1F6E
D680 - D6BF = 2036
D6C0 - D6FF = 1CB0
D700 - D73F = 1919
D740 - D77F = 164F
D780 - D7BF = 1ADC
D7C0 - D7FF = 19BC
D800 - D83F = 1A63
D840 - D87F = 19C7
D880 - D8BF = 1CB3
D8C0 - D8FF = 1DD0
```

```
260 LOCATE65,14:PRINTPEEK(&HE1D1):LINE(120,52)-(127,59),PRESET,5,BF:DEFUSR4=&HE4
16:AA=USR4(0)
270 COLOR3:LOCATE60,17:PRINT"MYSTERY POINT":LINE(126,72)-(137,79),PRESET,6,BF
280 COLOR5:LOCATE60,22:PRINT"S C E N E  01"
290 DEFUSR0=&HE37A:AA=USR0(0)
300 DEFUSR1=&HE130:AA=USR1(0)
310 F=PEEK(&HE9F8):IF F=1 THEN 320 ELSE GOTO 460
320 COLOR4:LOCATE15,10:PRINT"=================================="
330 LOCATE15,11:PRINT"**   G A M E   O V E R   **"
340 LOCATE15,12:PRINT"=========================="
350 FOR I=1TO3000:NEXT
360 GOTO 170
370 DEFUSR6=&HE166:AA=USR6(0)
380 COLOR6:LOCATE18,3:PRINT"P L A Y"
390 COLOR3:LOCATE10,5:PRINT"G A L A X I A N   G A M E"
400 COLOR4:LOCATE5,9:PRINT"K E Y  F U N C T I O N (ﾃﾝｷｰ)"
410 COLOR7:LOCATE19,11:PRINT"4 <=====> 6"
420 LOCATE5,13:PRINT"B E A M  ..........  S P A C E  K e y"
430 COLOR5:LOCATE5,17:PRINT"5000 POINT ｦ ｺｴﾙﾄ ﾋﾞｰﾑﾎｳ  1 ﾀﾞｲ ﾌｴﾏｽ｡"
440 COLOR7:LOCATE5,22:PRINT"H i t  R E T U R N  K e y  t o  S t a r t";
450 IF INPUT$(1)<>CHR$(13)THEN450 ELSE RETURN
460 COLOR4:LOCATE15,10:PRINT"=================================="
470 LOCATE15,11:PRINT"***   G I V E   U P    ***"
480 LOCATE15,12:PRINT"=========================="
490 COLOR6:LOCATE18,15:PRINT"Y O U   W I N   !!!!"
500 COLOR3:LOCATE17,17:PRINT"GALAXIAN ﾊ ｾﾞﾝﾒﾂ ｼﾏｼﾀ｡"
510 COLOR5:LOCATE18,20:PRINT"P L A Y   A G A I N ?"
520 LOCATE16,22:PRINT"H i t   R E T U R N  k e y ";
530 IF INPUT$(1)<>CHR$(13)THEN530 ELSE GOTO 170
5000 CLEAR300,&HD3FF:WIDTH40,25:CONSOLE0,25,0,0:PRINTCHR$(12)
5010 PRINT"GALAXIAN CHECK SUM"
5020 PRINT"START=D400   END=E495":PRINT:S=54272!:E=58517!:T=64
5030 X=0:R=S+T:IF R>=E THEN R=E+1
5040 FOR I=S TO R-1
5050 Y=PEEK(I):X=X+Y:NEXT
5060 K=S:GOSUB5110:S$=K$
5070 S=S+T:K=R-1:GOSUB5110:E$=K$
5080 K=X:GOSUB5110:X$=K$
5090 PRINTS$;" - ";E$;" = ";X$
5100 IF S>=E THEN END ELSE GOTO 5030
5110 U=INT(K/4096):K=K-4096*U:K$=HEX$(U)
5120 U=INT(K/256):K=K-256*U:K$=K$+HEX$(U)
5130 U=INT(K/16):K=K-16*U:K$=K$+HEX$(U)+HEX$(K)
5140 RETURN
```

```
D900 - D93F = 16A7
D940 - D97F = 1F25
D980 - D9BF = 0C44
D9C0 - D9FF = 12E8
DA00 - DA3F = 1701
DA40 - DA7F = 15B1
DA80 - DABF = 211B
DAC0 - DAFF = 1450
DB00 - DB3F = 19F0
DB40 - DB7F = 1F00
DB80 - DBBF = 18FD
DBC0 - DBFF = 1CE6
DC00 - DC3F = 1FDC
DC40 - DC7F = 1B8A
DC80 - DCBF = 16D3
DCC0 - DCFF = 1739
DD00 - DD3F = 142C
DD40 - DD7F = 197F
DD80 - DDBF = 183A
DDC0 - DDFF = 15F9
DE00 - DE3F = 14E4
DE40 - DE7F = 1D42
DE80 - DEBF = 1817
DEC0 - DEFF = 1AE1
DF00 - DF3F = 2588
DF40 - DF7F = 17A5
DF80 - DFBF = 1EB4
DFC0 - DFFF = 1F97
E000 - E03F = 1E4C
E040 - E07F = 1743
E080 - E0BF = 1927
E0C0 - E0FF = 15DD
E100 - E13F = 20EB
E140 - E17F = 1D0E
E180 - E1BF = 1956
E1C0 - E1FF = 1BEE
E200 - E23F = 1B8C
E240 - E27F = 154D
E280 - E2BF = 1073
E2C0 - E2FF = 1051
E300 - E33F = 12A4
E340 - E37F = 1603
E380 - E3BF = 2040
E3C0 - E3FF = 2274
E400 - E43F = 1981
E440 - E47F = 1996
E480 - E495 = 09DB
```

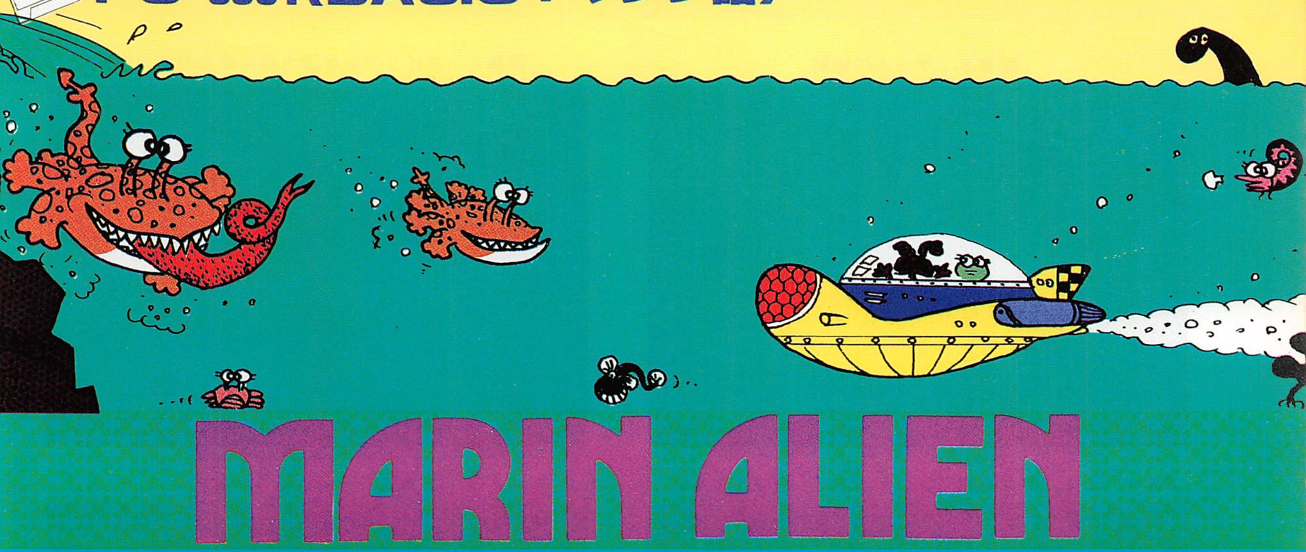
マシン語プログラム　ダンプ・リスト

```
D400: 36 42 23 36 EE 23 36 24 C9 00 00 00 36 23 23 36
D410: 66 23 36 32 C9 00 00 00 36 64 23 36 CC 23 36 46
D420: C9 00 00 00 36 31 23 36 66 23 36 13 C9 00 00 00
D430: 36 44 23 36 EE 23 36 44 C9 00 00 00 36 21 23 36
D440: 66 23 36 12 C9 00 00 00 36 30 23 36 EE 23 36 03
D450: C9 00 00 00 36 4F 23 36 D7 23 36 47 23 36 0F C9
D460: 00 00 00 06 09 CD 00 D4 23 23 10 F9 C9 00 00 00
D470: 21 62 F5 CD 63 D4 21 EA F4 CD 63 D4 21 72 F4 CD
D480: 63 D4 21 FE F3 06 07 CD 18 D4 23 23 10 F9 21 8A
D490: F3 06 05 CD 30 D4 23 23 10 F9 21 16 F3 CD 48 D4
D4A0: 21 1E F3 C3 48 D4 2A F5 E9 CD 54 D4 11 86 FF 19
D4B0: 36 E0 C9 00 00 00 00 00 00 00 01 10 27 CD D9 D4
D4C0: 01 E8 03 CD D9 D4 01 64 00 CD D9 D4 0E 0A CD D9
D4D0: D4 7D C6 30 12 C9 00 00 00 C5 A7 3E 30 ED 42 38
D4E0: 03 3C 18 F9 C1 09 12 13 C9 00 00 00 00 C5 3A F9
D4F0: E9 47 87 87 80 3C 32 F9 E9 C1 C9 00 00 00 00 00
D500: 16 05 1E 00 1B 7A B3 20 FB C9 00 00 00 21 FE E9
D510: 34 3A FD E9 3C FE 08 20 01 AF 32 FD E9 3A FF E9
D520: FE 01 C0 DB 09 FE FF C0 AF 32 FF E9 C9 00 00 00
D530: 3A FE E9 E6 07 C0 DD 21 FE E8 06 27 DD 7E 00 FE
D540: 01 28 0B 11 06 00 DD 19 10 F2 C9 00 00 00 DD 6E
D550: 01 DD 66 02 E5 36 00 23 36 00 23 36 00 E1 DD 7E
D560: 05 CB 67 20 1F FE 01 20 05 CD 0C D4 18 0C FE 02
D570: 20 05 CD 24 D4 18 03 CD 3C D4 DD CB 05 E6 C3 43
D580: D5 00 00 00 FE 11 20 05 CD 00 D4 18 0C FE 12 20
D590: 05 CD 18 D4 18 03 CD 30 D4 DD CB 05 A6 C3 43 D5
D5A0: 00 00 00 00 00 00 00 00 00 3A FE E9 E6 1F C0 DD
D5B0: 21 FE E8 06 29 DD 6E 01 DD 66 02 3A FC E9 FE 00
D5C0: 20 2F 23 DD 75 01 DD 74 02 DD 34 04 DD 7E 00 FE
D5D0: 01 C2 19 D6 DD 7E 04 FE 30 20 05 3E 01 32 F4 E9
D5E0: 0E 03 23 23 2B 7E 23 77 2B 0D 20 F8 18 29 00 00
D5F0: 00 2B DD 75 01 DD 74 02 DD 35 04 DD 7E 00 FE 01
D600: 20 17 DD 7E 04 FE 04 20 04 AF 32 F4 E9 0E 03 23
D610: 7E 2B 77 23 0D 20 F8 36 00 11 06 00 DD 19 05 C2
D620: B5 D5 3A F4 E9 32 FC E9 C9 00 00 00 00 00 00 00
D630: 00 00 00 00 00 3A FE E9 E6 03 C0 DB 00 FE EF CA
D640: 6E D6 FE BF C0 00 00 00 2A F5 E9 7D FE 70 C8 36
D650: 00 23 E5 22 F5 E9 CD 54 D4 E1 11 88 FF 19 36 00
D660: 3A FD E8 FE 00 C0 23 36 E0 C9 00 00 00 00 2A F5
D670: E9 7D FE 43 C8 2B 22 F5 E9 CD 54 D4 23 36 00 11
D680: 86 FF 19 36 00 3A FD E8 FE 00 C0 2B 36 E0 C9 00
D690: 00 00 00 00 00 3A FF E9 FE 00 C0 3A FD E8 FE 00
D6A0: C0 DB 09 FE BF C0 3E 01 32 FF E9 32 FD E8 CD C2
```

```
D6B0: D6 2A F5 E9 11 89 FF 19 22 FB E8 C9 00 00 00 00
D6C0: 00 00 06 06 3E 20 D3 40 16 01 CD 02 D5 AF D3 40
D6D0: 16 01 CD 02 D5 10 ED C9 00 00 00 00 3A FD E8 FE
D6E0: 01 C0 3A FE E9 E6 01 C0 2A FB E8 7E FE E0 20 02
D6F0: 36 00 11 88 FF 19 7C FE F2 20 08 AF 32 FD E8 C3
D700: D0 D9 00 22 FB E8 7E FE 42 28 1C FE 24 20 04 2B
D710: 2B 18 14 FE EE 20 19 2B 7E FE 42 28 0A FE 44 28
D720: 1D C3 64 D7 00 00 00 CD 46 D8 01 1E 00 C3 8D D8
D730: FE 44 20 16 2B 7E FE EE 20 03 2B 18 01 23 CD 46
D740: D8 01 32 00 C3 8D D8 00 00 00 FE 64 28 0A FE CC
D750: 28 05 FE 46 20 0E 2B 2B CD 46 D8 01 28 00 C3 8D
D760: D8 00 00 00 FE 23 28 1E FE 32 20 04 2B 2B 18 16
D770: FE 66 20 2B 2B 7E FE 23 28 0C FE 31 28 2B FE 21
D780: CA CC D7 C3 EC D7 CD 46 D8 01 1E 00 3A F8 E8 FE
D790: 01 28 06 CD 0A D9 01 3C 00 C3 8D D8 00 00 00 FE
D7A0: 31 28 06 FE 13 20 1B 2B 2B CD 46 D8 01 28 00 3A
D7B0: F8 E8 FE 01 28 06 CD 0A D9 01 50 00 C3 8D D8 00
D7C0: 00 00 FE 21 28 06 FE 12 20 22 2B 2B CD 46 D8 01
D7D0: 32 00 3A F8 E8 FE 01 28 0D CD 4C DF 3A F9 E8 3C
D7E0: 32 F9 E8 01 64 00 C3 8D D8 00 00 00 FE 30 28 13
D7F0: FE 03 20 04 2B 2B 18 0B FE EE C2 38 D8 2B 7E FE
D800: 30 20 35 DD 21 E8 E9 06 02 CD 4C D8 01 3C 00 3A
D810: F8 E8 FE 01 CA 8D D8 3A F9 E8 FE 00 20 05 01 C8
D820: 00 18 0F FE 01 20 05 01 90 01 18 06 FE 02 C0 01
D830: 20 03 CD 40 D9 C3 8D D8 FE 00 C0 36 E0 C9 00 00
D840: 00 00 00 00 00 00 DD 21 FE E8 06 27 DD 7E 00 FE
D850: 01 28 0F 11 06 00 DD 19 10 F2 AF 32 F8 E8 C9 00
D860: 00 00 EB DD 6E 01 DD 66 02 7C BA 20 04 7D BB 28
D870: 04 EB C3 53 D8 EB 3E 01 32 F8 E8 AF DD 77 00 3A
D880: FA E9 3D 32 FA E9 C9 00 00 00 00 00 00 36 A1 23
D890: 36 24 23 36 85 3E 20 D3 40 16 09 CD 02 D5 AF D3
D8A0: 40 36 00 2B 36 00 2B 36 00 2A F6 E8 09 22 F6 E8
D8B0: 11 87 F6 CD BA D4 CD C4 D8 AF 32 FD E8 C3 D0 D9
D8C0: 00 00 00 00 2A F6 E8 3A C8 E8 FE 00 C8 7C FE 13
D8D0: 20 07 A7 7D FE 88 D8 18 04 A7 FE 88 D8 3A FA E8
D8E0: 3C 32 FA E8 C6 30 11 D3 F9 12 06 0F 3E 20 D3 40
D8F0: 16 07 CD 02 D5 AF D3 40 16 01 CD 02 D5 10 ED AF
D900: 32 C8 E8 C9 00 00 00 00 00 00 DD 21 CA E8 06 04
D910: DD 7E 00 FE 01 28 0B 11 0A 00 DD 19 10 F2 C9 00
D920: 00 00 EB DD 6E 01 DD 66 02 7C BA 20 04 7D BB 28
D930: 04 EB C3 17 D9 EB C3 7B D8 00 00 00 00 00 00 00
D940: E5 3E 01 32 F3 E8 79 FE C8 20 08 21 B0 FB CD 79
D950: D9 18 18 FE 90 20 08 21 B0 FB CD 8B D9 18 0C FE
```

```
D960: 20 28 02 E1 C9 21 B0 FB CD 9D D9 21 B2 FB CD AF
D970: D9 E1 C3 4C DF 00 00 00 00 36 88 23 36 08 11 78
D980: 00 19 36 09 2B 36 AC C9 00 00 00 36 00 23 36 08
D990: 11 78 00 19 36 0F 2B 36 47 C9 00 00 00 36 88 23
D9A0: 36 08 11 78 00 19 36 0F 2B 36 AF C9 00 00 00 36
D9B0: 88 23 36 08 23 36 88 23 36 08 11 78 00 19 36 0F
D9C0: 2B 36 8F 2B 36 0F 2B 36 8F C9 00 00 00 00 00 00
D9D0: 2A F5 E9 11 89 FF 19 36 E0 C9 00 00 00 00 00 00
D9E0: D5 C5 21 00 00 11 78 00 06 08 CB 0F 30 01 19 EB
D9F0: 29 EB 05 20 F5 11 00 F3 19 C1 D1 16 00 19 C9 00
DA00: 00 00 00 00 00 00 00 00 00 00 21 C9 E8 3A FA E9
DA10: FE 1D 20 04 3E 02 77 C9 FE 0D 20 04 3E 03 77 C9
DA20: FE 06 C0 3E 04 77 C9 00 00 00 00 00 CD 0A DA 4E
DA30: DD 21 FE E8 FD 21 CA E8 06 22 DD 7E 00 FE 01 28
DA40: 0C 11 06 00 DD 19 10 F2 C9 00 00 00 00 FD 7E 00
DA50: FE 00 28 09 0D C8 11 0A 00 FD 19 18 F0 FD 36 00
DA60: 01 DD 6E 01 DD 66 02 E5 36 00 23 36 00 23 36 00
DA70: E1 A7 DD 7E 04 FE 1B 38 17 11 89 FF DD 7E 04 3C
DA80: E5 21 E8 DA FD 75 08 FD 74 09 E1 18 15 00 00 00
DA90: 11 87 FF DD 7E 04 3D E5 21 ED DA FD 75 08 FD 74
DAA0: 09 E1 FD 77 04 19 FD 75 01 FD 74 02 DD 7E 03 3D
DAB0: FD 77 03 DD 7E 05 FD 77 05 CD D3 DA DD E5 E1 FD
DAC0: 75 06 FD 74 07 AF DD 77 00 0D C8 C3 41 DA 00 00
DAD0: 3A CF E8 CB 47 20 03 C3 24 D4 CB 4F 20 03 C3 0C
DAE0: D4 C3 3C D4 00 00 00 00 02 01 01 00 00 FE FF FF
DAF0: 00 00 00 01 03 04 04 03 02 01 01 01 00 00 00 FF
DB00: FD FC FC FD FE FF FF FF 00 00 00 01 01 02 02 02
DB10: 01 01 00 FF FD FD 00 FF FF FE FE FE FF FF 00 01
DB20: 03 03 00 00 01 01 02 02 01 00 00 FF FF FF 00 00
DB30: FF FF FE FE FF 00 00 01 01 01 00 00 01 01 00 01
DB40: 00 01 00 01 00 00 00 00 00 FF FF 00 FF 00 FF FF
DB50: 00 00 00 00 00 00 3A FD E9 FE 01 C0 CD A0 DB CD
DB60: 80 DB CD A0 DB CD 80 DB CD 85 DB CD A0 DB CD 85
DB70: DB CD 8A DB CD A0 DB C3 8A DB 00 00 00 00 00 00
DB80: 21 D4 E8 18 08 21 DE E8 18 03 21 E8 E8 11 CA E8
DB90: 06 0A 4E 1A 77 79 12 23 13 10 F7 C9 00 00 00 00
DBA0: 3A CA E8 FE 01 C0 2A CB E8 36 00 23 36 00 23 36
DBB0: 00 3A CD E8 3C 32 CD E8 FE 06 20 05 CD 5D DD 18
DBC0: 07 FE 0E 20 03 CD 37 DC 3A CE E8 2A D2 E8 86 37
DBD0: 3F FE 34 30 02 18 0A AF 32 CA E8 C3 F0 DD 00 00
DBE0: 00 23 22 D2 E8 32 CE E8 5F 3A CD E8 F5 CD E0 D9
DBF0: F1 FE 0E 28 10 FE 10 28 0C 00 00 00 00 FE 12 20
DC00: 07 CB 4F 28 03 CD 86 DC FE 18 20 1C E5 06 03 7E
DC10: FE 4F 28 0C FE D7 28 08 FE 47 28 04 FE 0F 20 04
DC20: E1 C3 AC DC 23 10 E8 E1 3A CD E8 FE 19 CA D7 DB
DC30: 22 CB E8 C3 D0 DA 00 CD 41 DC 22 D2 E8 C9 00 00
DC40: 00 CD D0 DD FE 01 20 04 21 FE DA C9 FE 02 20 04
DC50: 21 16 DB C9 FE 03 20 04 21 46 DB C9 FE 04 20 04
DC60: 21 22 DB C9 FE 05 20 04 21 F2 DA C9 FE 06 20 04
DC70: 21 3A DB C9 FE 07 20 04 21 2E DB C9 21 0A DB C9
DC80: 00 00 00 00 00 00 23 DD 21 8C E8 06 14 DD 7E 00
DC90: FE 00 28 0B 11 03 00 DD 19 10 F2 C9 00 00 00 DD
DCA0: 36 00 01 DD 75 01 DD 74 02 C9 00 00 AF 32 CA E8
DCB0: 2A F5 E9 06 12 CD 07 DD 16 04 CD 47 DD CD 27 DD
DCC0: 16 06 CD 47 DD 10 EE 36 00 23 36 00 23 36 00 23
DCD0: 36 00 11 85 FF 19 36 00 23 36 00 23 36 00 23 36
DCE0: 00 3A FA E8 3D 32 FA E8 C6 30 11 D3 F9 12 3A FA
DCF0: E8 FE 00 20 06 3E 01 32 F8 E9 C9 3E 01 32 F7 E9
DD00: C3 7F D8 00 00 00 00 E5 36 80 23 36 92 23 36 AD
DD10: 23 36 49 11 85 FF 19 36 40 23 36 08 23 36 80 23
DD20: 36 80 E1 C9 00 00 00 E5 36 42 23 36 A9 23 36 AD
DD30: 23 36 24 11 85 FF 19 36 94 23 36 42 23 36 92 23
DD40: 36 24 E1 C9 00 00 00 3E 20 D3 40 CD 02 D5 AF D3
DD50: 40 16 04 CD 02 D5 C9 00 00 00 00 00 00 CD 67 DD
DD60: 22 D2 E8 C9 00 00 00 3A CE E8 FE 1B 38 2E CD D0
DD70: DD FE 01 28 08 FE 04 28 04 FE 06 20 04 21 FE DA
DD80: C9 FE 00 28 04 FE 05 20 04 21 16 DB C9 FE 02 20
DD90: 04 21 46 DB C9 21 2E DB C9 00 00 00 CD D0 DD FE
DDA0: 02 28 08 FE 05 28 04 FE 07 20 04 21 F2 DA C9 FE
DDB0: 03 28 04 FE 06 20 04 21 0A DB C9 FE 04 20 04 21
DDC0: 3A DB C9 21 22 DB C9 00 00 00 00 00 00 00 00 00
DDD0: 2A 89 E8 54 5D 19 19 7D 84 67 11 81 00 19 22 89
DDE0: E8 7C FE 08 D8 DE 08 18 F9 00 00 00 00 00 00 00
DDF0: DD 2A D0 E8 DD 36 00 01 DD 6E 01 DD 66 02 DD 7E
DE00: 04 FE 31 38 15 FE D0 30 14 2B 2B 2B 2B 3D 3D 3D
DE10: 3D FE 31 30 F4 18 16 00 00 00 FE 04 D0 23 23 23
DE20: 23 3C 3C 3C 3C FE D0 30 F4 FE 04 38 F0 DD 75 01
DE30: DD 74 02 DD 77 04 C9 00 00 00 00 00 00 00 00 00
DE40: DD 21 E8 E9 18 04 DD 21 EE E9 DD 36 00 00 3E 05
DE50: 32 70 E8 3E 01 32 6B E8 DD 6E 01 DD 66 02 22 6C
DE60: E8 DD 7E 03 32 6E E8 DD 22 71 E8 DD 7E 04 32 6F
DE70: E8 FE 1B 30 05 21 E8 DA 18 03 21 ED DA 22 73 E8
DE80: 22 7D E8 22 87 E8 C9 00 00 00 00 00 00 DD 21 CA
DE90: E9 18 04 DD 21 D6 E9 0E 02 06 03 FD 21 75 E8 DD
DEA0: 7E 00 FE 01 28 0B 11 06 00 DD 19 10 F2 C9 00 00
DEB0: 00 FD 77 00 DD 36 00 00 DD 7E 01 FD 77 01 DD 7E
DEC0: 02 FD 77 02 DD 7E 03 FD 77 03 DD 7E 04 FD 77 04
DED0: FD 36 05 03 DD E5 E1 FD 75 06 FD 74 07 0D C8 11
DEE0: 0A 00 FD 19 C3 A6 DE 00 00 00 00 00 00 DD 21 CA
DEF0: E9 06 05 0E 01 CD 9B DE FD 7E 04 C3 71 DE 00 00
DF00: 00 00 00 00 3A FA E9 FE 22 D0 3A 6B E8 FE 00 C0
DF10: 3A 75 E8 FE 00 C0 3A 7F E8 FE 00 C0 CD ED D4 FE
DF20: 6F C0 AF 32 F9 E8 CD 5A DF 3A E8 E9 FE 01 20 06
DF30: CD 40 DE C3 8D DE 3A EE E9 FE 01 C2 ED DE CD 46
DF40: DE C3 93 DE 00 00 00 00 00 00 00 00 C5 DD 21 6B
DF50: E8 06 03 CD 10 D9 C1 C9 00 00 CD 41 DC 22 67 E8
DF60: CD 41 DC 22 69 E8 C9 00 00 00 00 00 00 21 75 E8
DF70: 18 03 21 7F E8 11 6B E8 C3 90 DB 00 00 00 00 00
DF80: 3A FD E9 FE 04 C0 CD 6D DF CD A0 DF CD 6D DF CD
DF90: 72 DF CD A0 DF CD 72 DF C3 69 E0 00 00 00 00 00
DFA0: 3A 6B E8 FE 01 C0 2A 6C E8 36 00 23 36 00 23 36
DFB0: 00 3A 6E E8 3C 32 6E E8 FE 06 20 05 2A 67 E8 18
DFC0: 07 FE 0E 20 06 2A 69 E8 22 73 E8 3A 6F E8 2A 73
DFD0: E8 86 FE 34 38 0F AF 32 6B E8 DD 2A 71 E8 CD F4
DFE0: DD C3 4A E0 00 23 22 73 E8 32 6F E8 5F 3A 6E E8
DFF0: F5 CD E0 D9 F1 FE 0E 28 0C FE 10 28 08 FE 12 20
E000: 07 00 00 00 00 CD 86 DC FE 18 20 20 E5 06 03 7E
E010: FE 4F 28 0C FE D7 28 08 FE 47 28 04 FE 0F 20 08
E020: E1 AF 32 6B E8 C3 B0 DC 23 10 E4 E1 3A 6E E8 FE
E030: 19 CA D6 DF 22 6C E8 3A 70 E8 FE 03 CA 3C D4 C3
E040: 48 D4 00 00 00 00 00 00 00 00 3A E8 E9 FE 00 28
E050: 06 2A E9 E9 CD 48 D4 3A EE E9 FE 00 C8 2A EF E9
E060: C3 48 D4 00 00 00 00 00 00 21 67 E8 34 23 23 34
E070: CD A0 DF 21 67 E8 35 23 23 35 C9 00 00 00 00 00
E080: 3A FE E9 E6 03 C0 DD 21 8C E8 06 14 DD 7E 00 FE
E090: 01 28 0B 11 03 00 DD 19 10 F2 C9 00 00 00 DD 6E
E0A0: 01 DD 66 02 7E FE 06 20 02 36 00 11 78 00 19 7C
E0B0: FE FE 20 0F 7D FE B5 38 0A DD 36 00 00 C3 93 E0
E0C0: 00 00 00 DD 75 01 DD 74 02 7E FE 4F 28 0C FE D7
E0D0: 28 08 FE 47 28 04 FE 0F 20 0E DD 36 00 00 CD B0
E0E0: DC 21 FA E9 34 C9 00 00 FE 00 C2 93 E0 36 06 C3
E0F0: 93 E0 00 00 00 00 00 00 00 00 06 06 36 00 23 10
E100: FB C9 00 00 00 3A F3 E8 FE 00 C8 3A F2 E8 3D 32
E110: F2 E8 FE 00 C0 AF 32 F3 E8 3E 80 32 F2 E8 21 B0
E120: FB CD FA E0 21 28 FC C3 FA E0 00 00 00 00 00 00
E130: 3A F7 E8 47 3A F5 E8 B8 28 06 37 3F 90 D0 18 0B
E140: 3A F6 E8 4F 3A F4 E8 37 3F 91 D0 3A F7 E8 32 F5
E150: E8 3A F6 E8 32 F4 E8 2A F4 E8 11 67 F8 C3 BA D4
E160: 00 00 00 00 00 00 21 89 F2 11 78 00 0E 19 19 E5
E170: AF 06 36 77 23 10 FC E1 0D 20 F3 C9 00 00 00 00
E180: 3A FA E9 FE 01 C0 21 CA E8 06 04 7E FE 01 C8 11
E190: 0A 00 19 10 F6 21 6B E8 06 03 7E FE 01 C8 19 10
E1A0: F9 AF 32 FA E9 C9 00 00 00 00 AF 32 FD E9 32 F4
E1B0: E8 32 F5 E8 C9 00 00 00 00 00 00 00 00 00 00 00
E1C0: AF 32 FF E9 32 F8 E9 00 00 00 32 F6 E8 32 F7 E8
E1D0: 3E 04 32 FA E8 3E 01 32 C8 E8 32 8B E8 18 09 00
E1E0: 00 00 00 00 00 00 00 00 CD ED D4 E6 01 20 03 AF
E1F0: 18 02 3E 01 32 FC E9 AF 32 FB E9 32 F7 E9 32 F4
E200: E9 3E 29 32 FA E9 21 5A FE 22 F5 E9 21 5A E2 11
E210: FE E8 01 F6 00 ED B0 AF 32 F3 E8 3E 80 32 F2 E8
E220: 21 CA E8 11 0A 00 06 04 36 00 19 10 FB 3E 01 32
E230: C9 E8 21 8C E8 11 03 00 06 14 36 00 19 10 FB AF
E240: 32 6B E8 32 75 E8 32 7F E8 3E F0 32 66 E8 AF 32
E250: FD E8 C9 00 00 00 00 00 00 00 01 92 F4 03 2A 11
E260: 01 72 F4 03 0A 11 01 16 F4 02 26 12 01 FE F3 02
E270: 0E 12 01 0A F5 04 2A 11 01 EA F4 04 0A 11 01 82
E280: F5 05 2A 11 01 62 F5 05 0A 11 01 8E F4 03 26 11
E290: 01 76 F4 03 0E 11 01 06 F5 04 26 11 01 EE F4 04
E2A0: 0E 11 01 7E F5 05 26 11 01 66 F5 05 0E 11 01 12
E2B0: F4 02 22 12 01 02 F4 02 12 12 01 8A F4 03 22 11
E2C0: 01 7A F4 03 12 11 01 02 F5 04 22 11 01 F2 F4 04
E2D0: 12 11 01 7A F5 05 22 11 01 6A F5 05 12 11 01 0E
E2E0: F4 02 1E 12 01 06 F4 02 16 12 01 0A F4 02 1A 12
E2F0: 01 86 F4 03 1E 11 01 82 F4 03 1A 11 01 7E F4 03
E300: 16 11 01 F6 F4 04 16 11 01 FA F4 04 1A 11 01 FE
E310: F4 04 1E 11 01 76 F5 05 1E 11 01 72 F5 05 1A 11
E320: 01 6E F5 05 16 11 01 8A F3 01 12 13 01 8E F3 01
E330: 16 13 01 92 F3 01 1A 13 01 96 F3 01 1E 13 01 9A
E340: F3 01 22 13 01 16 F3 00 16 00 01 1E F3 00 1E 00
E350: 00 00 00 00 00 00 00 00 00 00 3E F0 32 66 E8 AF
E360: 32 F7 E9 32 FD E8 2A FB E8 36 00 21 5A FE 22 F5
E370: E9 C9 00 00 00 00 00 00 00 00 CD C0 E1 CD 70 D4
E380: CD 0D D5 16 05 CD 02 D5 CD 30 D5 CD A9 D5 3A 66
E390: E8 FE 02 20 03 CD A6 D4 FE 00 28 09 3D 32 66 E8
E3A0: 18 19 00 00 00 CD 35 D6 CD 95 D6 CD DC D6 3A FE
E3B0: E9 E6 1F 20 03 CD 2C DA CD 04 DF CD 56 DB CD 80
E3C0: DF CD 80 E0 CD 05 E1 3A F8 E9 FE 01 C8 3A F7 E9
E3D0: FE 01 20 09 CD 5A E3 C3 80 E3 00 00 00 CD 80 E1
E3E0: 3A FA E9 FE 00 C2 80 E3 3A 8B E8 3C 32 8B E8 FE
E3F0: 1A 20 06 3E 05 32 F8 E9 C9 6F 26 00 11 98 FD CD
E400: CC D4 CD 66 E1 CD E8 E1 16 FF CD 02 D5 C3 7D E3
E410: 00 00 00 00 00 00 21 CD F9 C3 A9 D4 00 00 00 00
E420: 11 78 00 77 19 10 FC C9 00 00 00 3E 18 06 19 21
E430: 53 F3 CD 20 E4 06 19 21 55 F3 CD 20 E4 06 19 21
E440: 59 F3 CD 20 E4 06 19 21 5B F3 CD 20 E4 06 19 21
E450: 51 F3 CD 20 E4 00 00 21 BF F4 3E 38 06 03 CD 20
E460: E4 3E 98 06 04 CD 20 E4 3E 78 06 04 CD 20 E4 3E
E470: D8 06 05 CD 20 E4 3E 58 06 04 CD 20 E4 3E B8 06
E480: 02 CD 20 E4 3E 78 32 47 F4 3E D8 32 57 F3 3E 58
E490: 32 CF F3 C9 00 00
```

PC-8001(BASIC+マシン語)



MARIN ALIEN

マリン・エイリアン　芸夢狂人

インベーダーに蹂躙された地球防衛軍は，もうゲーム・センターの片隅に残るのみとなったが（ちなみに筆者は6,000点の壁が越えられなかった），ここ海底の秘密基地にある地球防衛海軍はいまだ健在であった．

しかし，ここもエイリアンに発見され，いまやその防衛はサブマリン部隊の負うところとなった．空中から大挙して押し寄せるエイリアンはミサイルを落とし，海中には浮遊機雷も漂っている．さらに海面に近づいたエイリアンは海中へ入って体当りをかけてくる．

あなたは，これら障害を避けながらすべてのエイリアンをビーム砲で破壊しなくてはならない．地球の運命は今やあなたの手に握られた……！

遊び方

画面の右上から，SCORE，HI-SCORE，サブマリン数，SCENE（局面の数）が表示されます．

サブマリンは最初3台で，4,000点を越えると1台追加されます．サブマリンは上，下，左，右に動き，テンキーの[2]，[4]，[6]，[8]のキーで動きます（図1）．サブマリンが後へ動くというのも少し変ですが，あまり気にしないでください．

[スペース]キーを押すとビームが発射されます．キーをいったん離さないと次のビームは出ません．また，ビームが消えるまで，次のビームは出ません．空中のエイリアンはバリアを張っているので，中央に当らないと破壊されません．海中ではバリアが張れないので，どこに当っても破壊できます．

図1　サブマリンの移動

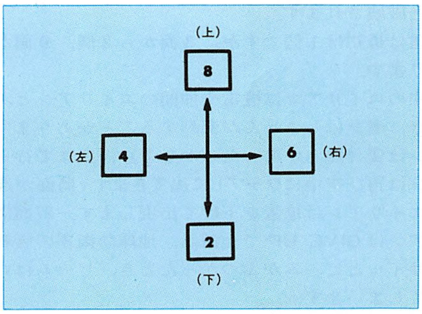


写真1　エイリアンが海にもぐると，斜めに移動する．

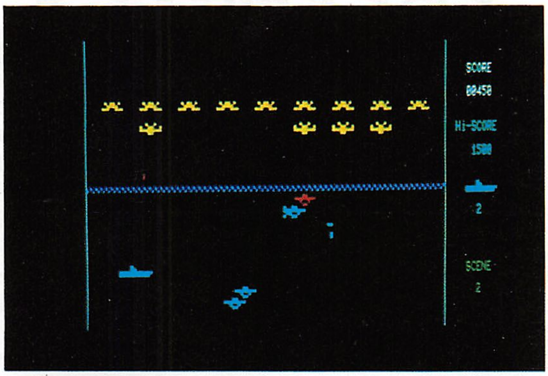


写真2　浮遊機雷にぶつかり，潜水艦が爆破される．

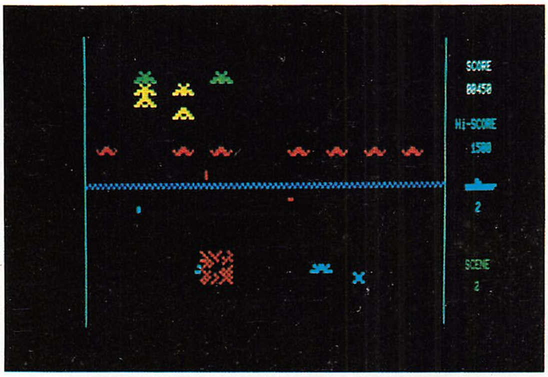


写真3　ゲームの説明

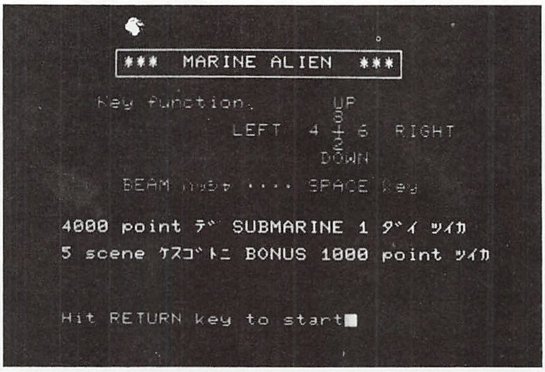
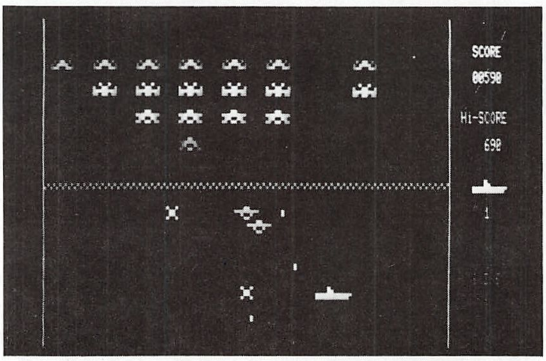


写真4　エイリアンを迎え撃つか,それとも逃げまわるか…



サブマリンがエイリアン，浮遊機雷(▣)，ミサイルに当たると破壊されます．

機雷は最初は１個ですが，３面から２個，９面から３個になります．

海中のエイリアンは機雷や仲間のエイリアンとぶつかったとき，爆発はしませんが変形することがあります（エイリアンは変身能力があったのかな？)．海底まで行ったエイリアンは再び空中にワープして出てきます．局面が進むにつれ，エイリアンは位置を下げて出現します．27面消すとエイリアンはGIVE UPで全滅し，地球防衛軍の勝利です．

ミサイルとビームがぶつかったとき，ビームはいつも消滅してしまいます．

得点

エイリアンの得点は表１のようになっています．１行目というのは一番下の行で，４行目というのは一番上の行です（海中のエイリアンを狙った方が高得点ですが‥‥危険も一杯！)．

機雷は撃っても得点にはなりません．５面ごとにＢＯＮＵＳが 1,000点つきます．

表１　エイリアンの得点

エイリアン	空中	海中
１行目，２行目	30	50
３行目	50	100
４行目	70	150

写真5　エイリアンが潜水艦に体当りする直前

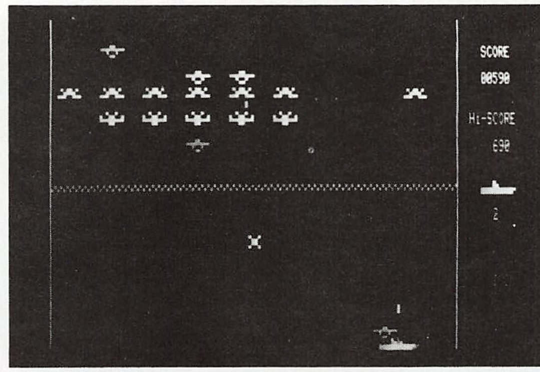
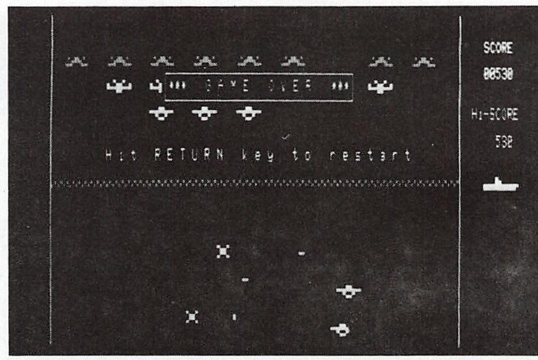


写真6　ゲームオーバー



高得点のコツ

機雷は左からしか出ないし，海底に着いたエイリアンが空中へワープするのも左側に出る確率が高いのです．まず，右側のエイリアンを全部消して安全域を作っておきます．そして，あとは海底まで潜って，無理をしないように海中のエイリアンを消していきます．

海中では左側のエイリアンは右下へ，右側のエイリアンは左下へ移動します．慣れてくるとどこへ降りてくるかわかるので，そこで待っています．ミサイルは，かたまって落ちてくることが多いので注意してください．

プログラムについて

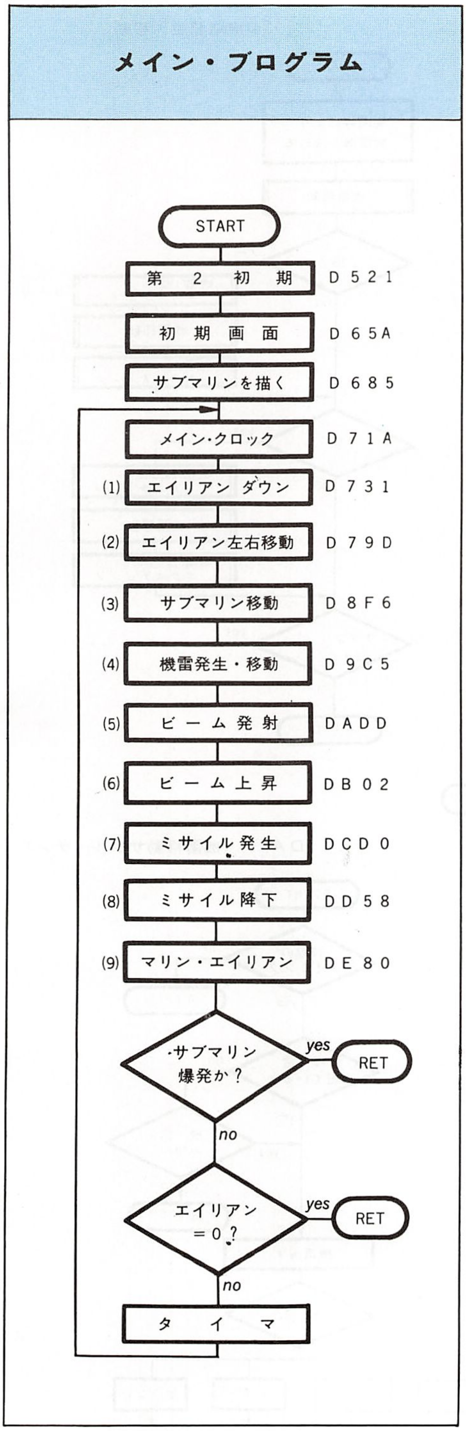
（１）マシン語

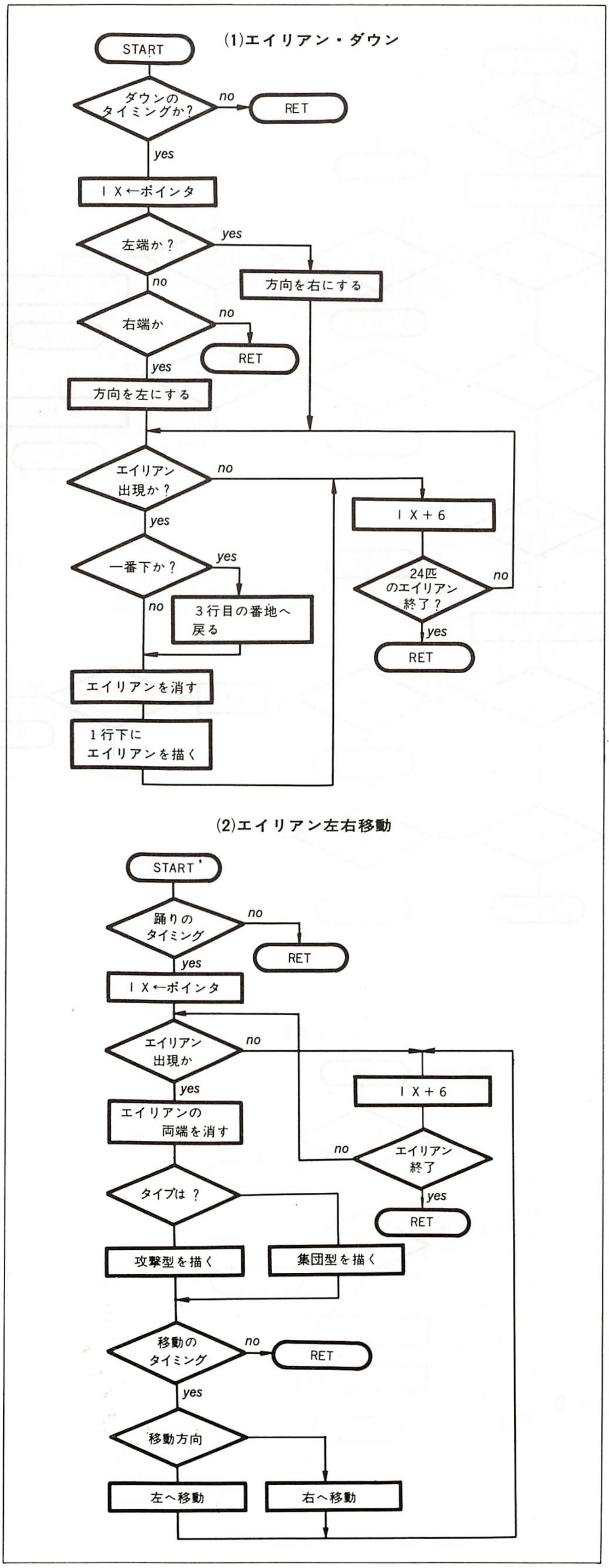
動きを必要とする重要な部分はすべてマシン語で組んでいます．そのため，サブマリンの爆発時やエイリアン＝0のときと終了時しか BASIC に戻らないので，それ以外は **STOP** キーがききません．途中で止めたいときは，**STOP** キーを押してから **RESET** キーを押すようにしてください．

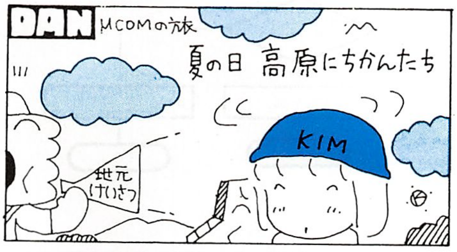
プログラムを入力したら，RUNする前に必ずテープにとっておきましょう．マシン語は１つ間違えただけでも暴走する危険があります．

マシン語については，かなり詳しくフローチャートを書

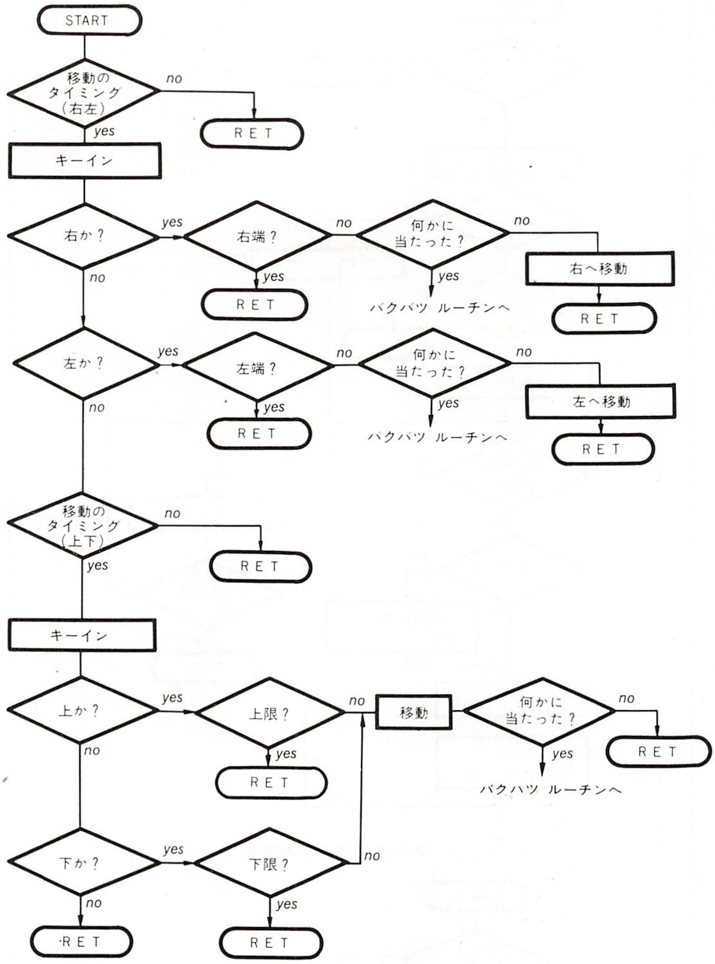
図.2


メイン・プログラム
START
第 2 初 期 D521
初期画面 D65A
サブマリンを描く D685
メイン・クロック D71A
(1) エイリアン ダウン D731
(2) エイリアン左右移動 D79D
(3) サブマリン移動 D8F6
(4) 機雷発生・移動 D9C5
(5) ビーム発射 DADD
(6) ビーム上昇 DB02
(7) ミサイル発生 DCD0
(8) ミサイル降下 DD58
(9) マリン・エイリアン DE80
サブマリン爆発か？
yes
RET
no
エイリアン＝0？
yes
RET
no
タ イ マ

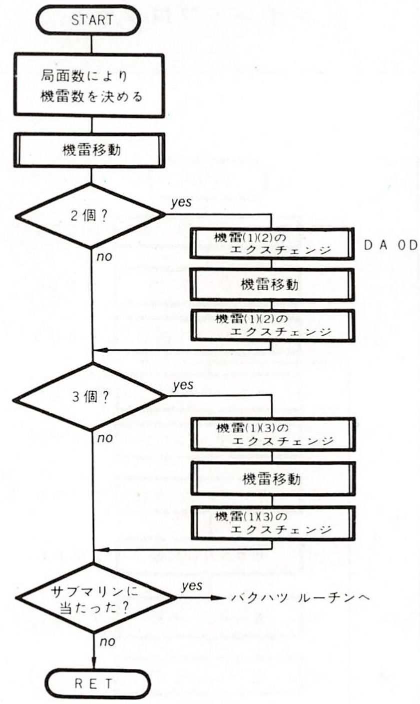

(1)エイリアン・ダウン
START
ダウンのタイミングか？
no
RET
yes
IX←ポインタ
左端か？
yes
方向を右にする
no
右端か
no
RET
yes
方向を左にする
エイリアン出現か？
no
IX＋6
24匹のエイリアン終了？
no
yes
RET
yes
一番下か？
yes
3行目の番地へ戻る
no
エイリアンを消す
1行下にエイリアンを描く
(2)エイリアン左右移動
START
踊りのタイミング
no
RET
yes
IX←ポインタ
エイリアン出現か
no
IX＋6
エイリアン終了
no
yes
RET
yes
エイリアンの両端を消す
タイプは？
攻撃型を描く
集団型を描く
移動のタイミング
no
RET
yes
移動方向
左へ移動
右へ移動


DAN
μCOMの旅
夏の日 高原にちかんたち
KIM
地元けいさつ

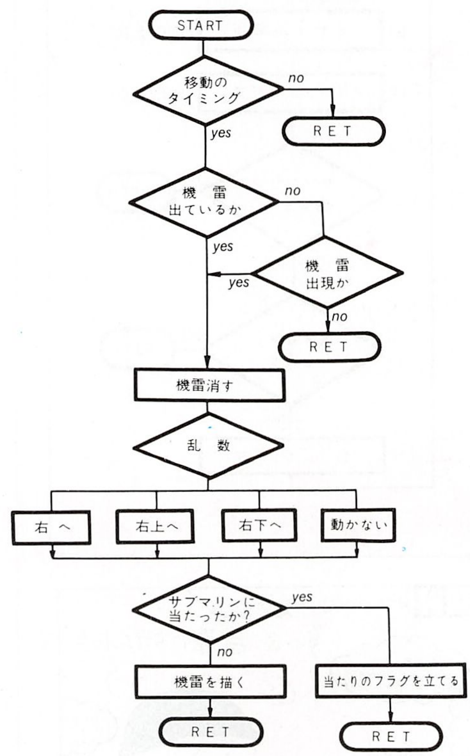
(3)サブマリン移動



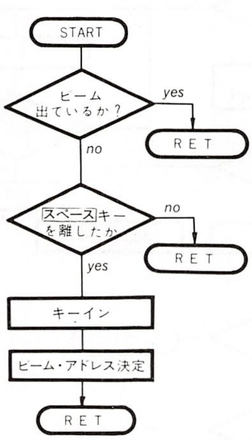
(4)機電発生・移動



DA28 〈機雷移動サブルーチン〉

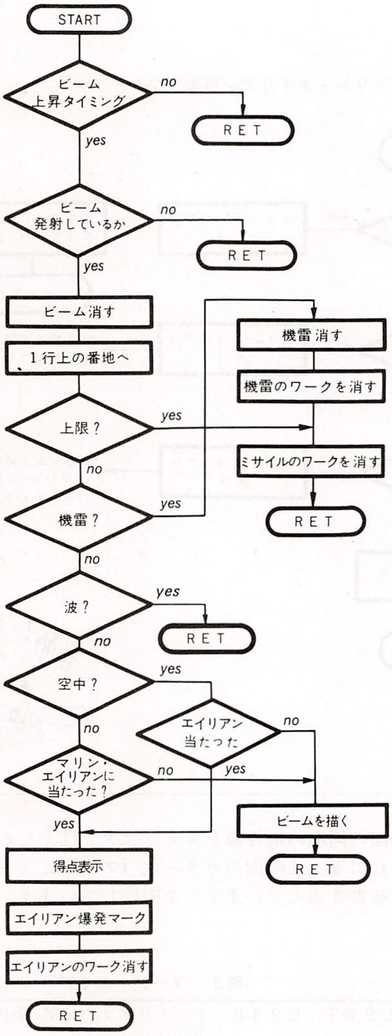


(5)ビーム発射

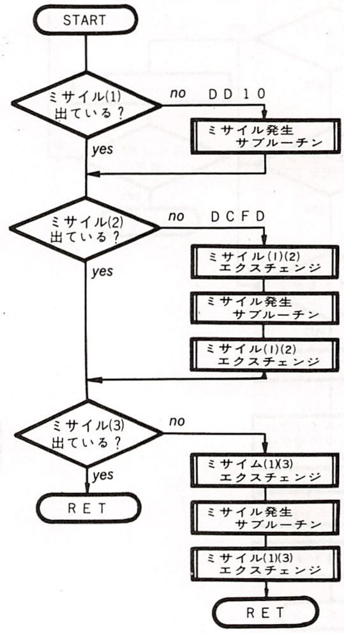




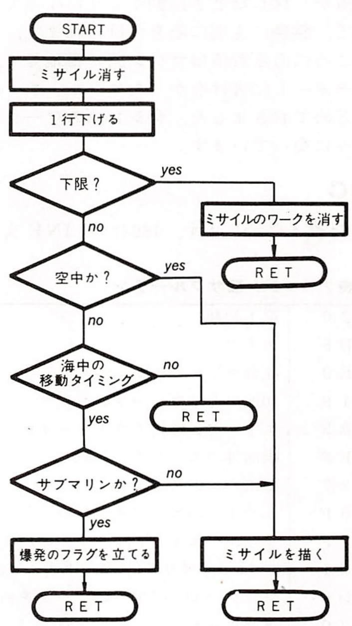
(6)ビーム上昇



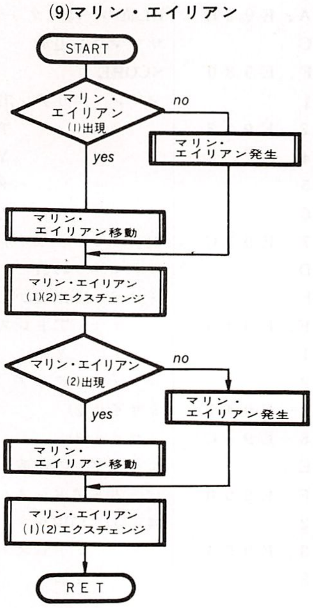
(7)ミサイル発生

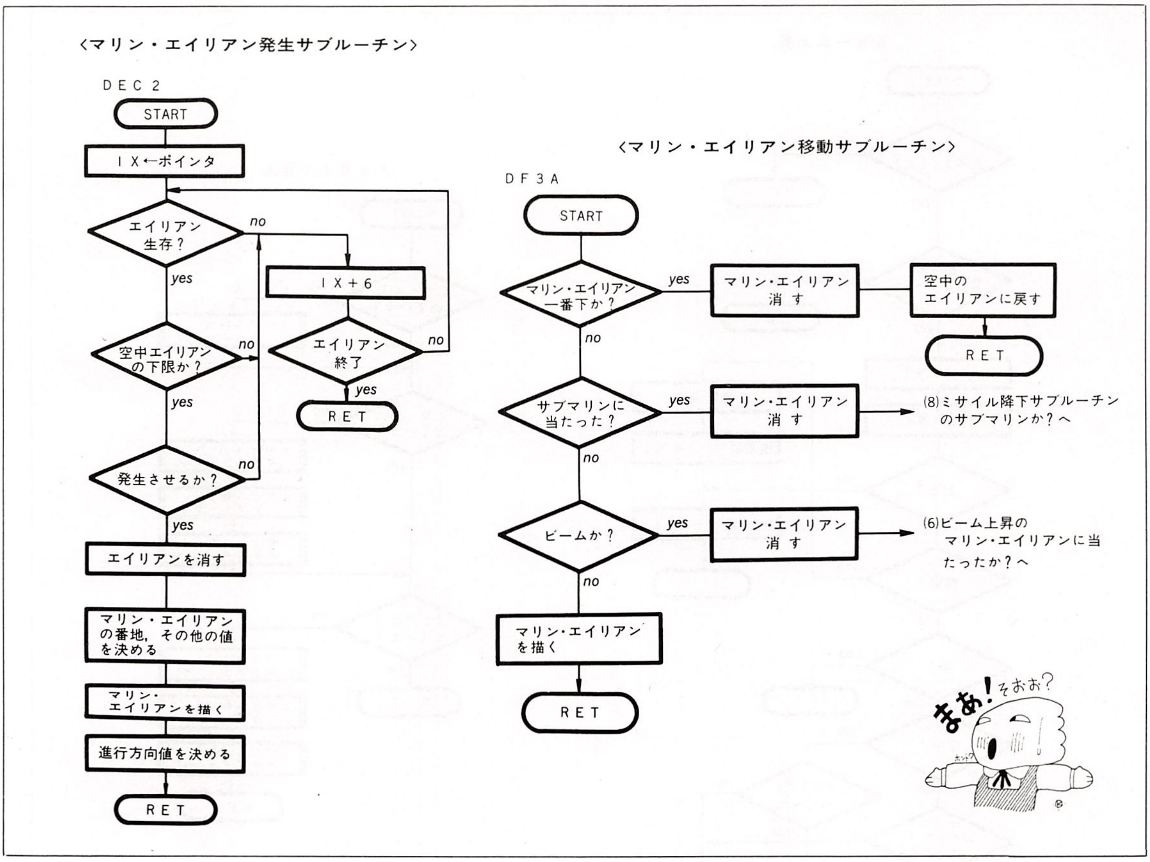


(8)ミサイル降下サブルーチン



(9)マリン・エイリアン





きました(図2)．これを見ればだいたいの動きはわかると思います．各ブロックはサブルーチンにしてあり，メイン・クロックの数値が一致したときに動くようになっています．これによって，容易に速度に差をつけられます．フローチャートのところにある数値はマシン語の番地です．

また，フローチャートに書けなかったサブルーチンについては**表2**にまとめておきました．マシン語のワーク・エリアは**表3**のようになっています．

(2) BASIC

BASICの説明は**表4**に示します．160行の**LINE**文は，画面の両端の境界線をキャラクタで描くためにあります．これによって画面のカラー化（アトリビュート・エリアに直接書き込んでいます）を楽にしています．

表2　マシン語サブルーチン

D500～D520	第1初期
D6CC～D6D8	タイマ
D6D9～D6E9	乱数サブルーチン
D6EA～D719	10進出力サブルーチン
D863～D8BF	エイリアンを書くサブルーチン
D8C0～D8F5	画面外のエイリアン移動
DA0D～DA27	機雷のエクスチェンジ
DCFD～DD0F	ミサイルのエクスチェンジ
DE16～DE7F	マリン・エイリアンを空中に戻す
DEAC～DEC1	マリン・エイリアンのエクスチェンジ
DFB0～DFDF	マリン・エイリアンのワークを消す
E020～E029	爆発ルーチン
E02A～E04F	音出しのサブルーチン
E050～E0E2	第3初期
E0E3～E0F0	BONUS処理
E0F1～E04C	カラー化

表3　ワーク・エリア

E927，E928	エイリアンの位置（開始時）
E929	乱数値
E92A，E92B	HL値の一時スタック
E92C	サブマリン追加
E92F，E930	SCORE
E931	マリン・エリアン出現か
E932，E933	〃　アドレス
E934	〃　Y値
E935	〃　タイプ
E936	〃　方向値
E937～E93C	2台目のマリン・エイリアン……上と同様
E93D	サブマリンの数
E93E	ミサイル(1)出現か
E93F，E940	〃　アドレス
E941	〃　Y値
E942	〃　タイプ
E943～E947	ミサイル(2)
E948～E94C	ミサイル(3)
E94E	エイリアンの位置　Y
E94F，E950	ビーム・アドレス
E952	機雷(1)　出現か
E953，E954	〃　アドレス
E955	〃　Y値

E956	機雷(1)X値
E957～E95B	〃(2)
E95C～E960	〃(3)
E962	サブマリンに当たったか
E963	機雷の数
E964	局面数
E965	何かにぶつかったか
E966	サブマリン・アドレス　Y値
E967	〃　X値
E968, E969	〃
E96A	エイリアンの数
E96B	エイリアン　X値
E96C	エイリアン(1)出現か
E96D, E96E	〃　アドレス
E96F	〃　Y値
E970	〃　X値
E971	〃　タイプ
E972～E9FB	エイリアン(2)～エイリアン(24)　上と同様
E9FD	メイン・クロック値
E9FE	エイリアン移動方向
E9FF	スペースキーを離したか

表4　BASICプログラム

130～160	初期設定
170	カラー化
180～210	初期画面
220	マシン語メイン・プログラムへ
260～320	局面数＋1，BONUS処理
340～540	爆発の画面
560～600	GIVE UP
620～690	GAME OVER
700～710	SCOREをSCに入れる．
720～810	爆発のデータ
840～950	説明文

変更点

うまくなって『点がとれすぎてつまらない』という人や，反対に『ちっとも点がとれなくてつまらない』という人のために，**表5**に変更できる項目を示します．

表5　変更点

変更する項目	番　地	現在の数値
(1)全体のスピード	E017	09
(2)サブマリンの数	D516	03
*(3)エイリアン移動速度	D736 D817 D8D4	0F
(4)エイリアンの踊り	D7A1	05
*(5)サブマリンの移動　左右	D8FA	03
上下	D90C	07
*(6)機雷の速度	DA2C	0F
*(7)ミサイルの速度　空中	DD5C	03
海中	DDCF	07
*(8)マリン・エイリアンの速度	DE84	03
(9)GIVE UPの局面数	BASICで行番号320	

(注) (3)は3ヶ所の値を同じにすること．

番号の頭に＊のついたものでは，変更する数値は2の倍数，つまり01，03，07，0F，1F，3F‥‥などにしないと動きがおかしくなります．速度は数値を小さくすれば速く，大きくすれば遅くなります．

(4)のエイリアンの踊りは，ぜひ数値を変えてみてください．いろいろと面白い動き方をします．(2)のサブマリンの数は，10台以上では表示がASCIIコードになることがあります．(9)のGIVE UPの局面数はBASICで，320行のA＝28の数値を変更します．

グリーン・モニタしかない人へ

プログラムはカラー用のため，グリーン・モニタでは，青，赤，紫の色はほとんど見えません．もっともミサイルやエイリアンが消えてから突然出てきたりしてよりスリルが味わえるという利点(？)もありますが‥‥．

変更は，BASIC文中の**COLOR**文の数値を7に変え，行番号170を削除してください．

あとがき

某日，授業が早く終わったので久しぶりにゲーム・センターへ行ったところ，『トマホーク777』というゲームがありました．なかなかおもしろく，とくに海中でエイリアンが襲ってくるのに感激して，さっそく作ってみたのがこのゲームです．

フローチャートを書くのに1週間かかり，ハンド・アセンブルの悲しさで，マシン語に直すのにまた1週間かかってしまいました．

マシン語を始めてまだ間がないので，かなり無駄なことをしている部分もあるみたいですが，それでもマシン語のスピードはスバラシイもので，BASICなど足もとにもおよびません．BASICのゲームの遅さにガッカリしている方は，ぜひマシン語をやってみることをお推めします．

最初は画面にキャラクタやドットを書く部分からマシン語にしていくと，とっつきやすいと思います．とにかくいじっていると自然にいろいろなことがわかってきます．ゲームの命は何といってもスピード……だと思います．

このゲームで筆者の最高点は16面で23,000点でした．

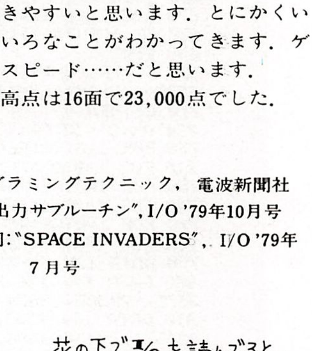
◘参考文献

1）Z80マイコンプログラミングテクニック，電波新聞社

2）八丸鉢真留："10進出力サブルーチン", I/O '79年10月号

3）近藤洋一，近藤康司："SPACE INVADERS", I/O '79年7月号

4）平安京エイリアン



```
90 'MARINE ALIEN   "MA"   VER 1.5   S55,6,6
100 CLEAR300,&HD4FF
110 DIMA%(2),B%(8),C%(5),D%(9),E%(6),F%(9),G%(6)
120 GOSUB720:GOSUB820
130 POKE&HE9FE,0
140 DEFUSR0=&HD500:AA=USR0(0):A=1:SC=0
150 WIDTH80,25:CONSOLE0,25,0,1:COLOR7,0,1:PRINTCHR$(12)
160 LINE(0,0)-(0,24)," ",B:LINE(67,0)-(67,24)," ",B
170 DEFUSR7=&HE0F1:AA=USR7(0)
180 COLOR7:LOCATE71,2:PRINT"SCORE":LOCATE71,4:PRINTUSING"#####";SC
190 COLOR5:LOCATE69,7:PRINT"Hi-SCORE":LOCATE71,9:PRINTUSING"#####";HS
200 LOCATE72,14:PRINTPEEK(&HE93D)
210 COLOR4:LOCATE71,19:PRINT"SCENE":LOCATE72,21:PRINTA
220 DEFUSR1=&HDFE3:AA=USR1(0)
230 IF PEEK(&HE96A)=0 THEN 240 ELSE GOTO 340
240 FOR I=1TO15:BEEP1:FOR K=1TO50:NEXT:BEEP0:FOR J=1TO50:NEXTJ,I
250 FOR I=1TO1000:NEXT
260 A=A+1:POKE&HE964,A:GOSUB700
270 IF A MOD5=0 THENSC=SC+1000 ELSE GOTO 320
280 DEFUSR8=&HE0E3:AA=USR8(0)
290 PRINTCHR$(12):FOR I=50TO1000STEP50:BEEP1
300 COLOR6:LOCATE30,11:PRINT"B O N Ú S  ";:PRINTUSING"####";I:FOR U=1TO30:NEXT
310 LOCATE30,11:PRINT"                              ":BEEP0:NEXT:LOCATE30,11:PRINT"B O N U
S  1000":FOR T=1TO800:NEXT
320 IF A=28 THEN 550
330 GOTO 150
340 DEFUSR9=&HE02A:X=PEEK(&HE967):Y=PEEK(&HE966):IF Y=24 THEN440
350 FOR K=1TO2
360 COLOR6:PUT@(X*2+2,Y*4)-(X*2+9,Y*4+3),A%,PSET
370 HL=5841:AA=USR9(HL)
380 COLOR4:PUT@(X*2+2,Y*4-4)-(X*2+11,Y*4+7),B%,PSET
390 HL=9128:AA=USR9(HL)
400 COLOR3:PUT@(X*2,Y*4-4)-(X*2+11,Y*4+7),D%,PSET
410 HL=9682:AA=USR9(HL)
420 COLOR2:PUT@(X*2,Y*4-4)-(X*2+11,Y*4+7),F%,PSET
430 NEXT:FOR I=1TO800:NEXT:GOTO 530
440 FOR K=1TO2
450 COLOR6:PUT@(X*2+2,Y*4)-(X*2+9,Y*4+3),A%,PSET
460 HL=5841:AA=USR9(HL)
470 COLOR4:PUT@(X*2+2,Y*4-4)-(X*2+11,Y*4+3),C%,PSET
480 HL=9128:AA=USR9(HL)
490 COLOR3:PUT@(X*2,Y*4-4)-(X*2+11,Y*4+3),E%,PSET
500 HL=9682:AA=USR9(HL)
510 COLOR2:PUT@(X*2,Y*4-4)-(X*2+11,Y*4+3),G%,PSET
520 NEXT:FOR I=1TO800:NEXT
530 S=PEEK(&HE93D):S=S-1:LOCATE73,14:PRINTS:IF S=0 THEN610 ELSE POKE&HE93D,S
540 GOTO 260
550 COLOR6:LOCATE21,4:PRINT"┌────────────────────────┐":BEEP
560 LOCATE21,5:PRINT"│###   G I V E   U P   ###│":BEEP
570 LOCATE21,6:PRINT"└────────────────────────┘":BEEP
580 COLOR5:LOCATE15,9:PRINT"ﾁｷｭｳ ﾎﾞｳｴｲｸﾞﾝ ﾉ ｼｮｳﾘ ﾃﾞｽ｡"
590 LOCATE15,10:PRINT"ALIEN ﾊ ｾﾞﾝﾒﾂ ｼﾏｼﾀ｡"
600 FOR I=1TO3000:NEXT
610 COLOR6:LOCATE19,4:PRINT"┌──────────────────────────┐"
620 LOCATE19,5:PRINT"│***   G A M E   O V E R   ***│"
630 LOCATE19,6:PRINT"└──────────────────────────┘"
640 GOSUB700
650 LOCATE15,9:PRINT"                              "
660 COLOR5:IF SC>HS THEN HS=SC:LOCATE71,9:PRINTUSING"#####";HS
670 LOCATE10,10:PRINT"H i t   R E T U R N   k e y   t o   r e s t a r t  ";
680 IF INPUT$(1)<>CHR$(13)THEN680 ELSE GOTO 130
690 END
700 SC=VAL(CHR$(PEEK(&HF528))+CHR$(PEEK(&HF529))+CHR$(PEEK(&HF52A))+CHR$(PEEK(&H
F52B))+CHR$(PEEK(&HF52C)))
710 RETURN
720 A%(0)=32:A%(1)=10308:A%(2)=-8724
730 FOR I=0TO8:READ J:B%(I)=J:NEXT
740 FOR I=0TO5:READ J:C%(I)=J:NEXT
750 FOR I=0TO9:READ J:D%(I)=J:NEXT
760 FOR I=0TO6:READ J:E%(I)=J:NEXT
770 FOR I=0TO9:READ J:F%(I)=J:NEXT
780 FOR I=0TO6:READ J:G%(I)=J:NEXT:RETURN
790 DATA 120,0,-32768,17700,16552,10389,16457,2,0,80,0,-32768,17700,16552,10389
800 DATA 144,136,8197,-23375,-24300,20874,-22363,-31452,1188,2337,96,136,8197,-2
3375,-24300,20874,-22363
810 DATA 144,9353,9349,-23371,-24300,384,-22363,-23260,9636,-30427,96,9353,9349,
-23371,-24300,384,-22363
820 WIDTH40,25:CONSOLE0,25,0,1:PRINTCHR$(12)
830 COLOR3:LOCATE4,2:PRINT"┌────────────────────────┐"
840 LOCATE4,3:PRINT"│***  MARINE ALIEN   ***│"
850 LOCATE4,4:PRINT"└────────────────────────┘"
860 COLOR4:LOCATE3,6:PRINT"Key function        UP"
870 LOCATE22,7:PRINT"8"
880 LOCATE14,8:PRINT"LEFT   4 ┼ 6   RIGHT"
890 LOCATE22,9:PRINT"2"
```

```
900 LOCATE21,10:PRINT"DOWN"
910 LOCATE5,12:PRINT"BEAM ﾊｯｼｬ ････ SPACE key"
920 COLOR7:LOCATE0,15:PRINT"4000 Point ﾃﾞ SUBMARINE 1 ﾀﾞｲ ﾂｲｶ"
930 LOCATE0,17:PRINT"5 scene ｸｽｺﾞﾄﾆ BONUS 1000 Point ﾂｲｶ"
940 COLOR5:LOCATE0,22:PRINT"Hit RETURN key to start";
950 IF INPUT$(1)<>CHR$(13)THEN950 ELSE RETURN
```

マシン語プログラム　ダンプ・リスト

```
D500: 3E 01 32 64 E9 32 63 E9 AF 32 4E E9 32 2C E9 21
D510: 00 00 22 2F E9 3E 03 32 3D E9 C9 00 00 00 00 00
D520: 00 AF 32 FF E9 21 6C E9 3E 01 CD 43 D6 21 70 E9
D530: 3E 03 CD 43 D6 21 71 E9 3E 01 06 08 CD 45 D6 21
D540: A1 E9 3E 02 06 07 CD 45 D6 21 CB E9 3E 03 06 09
D550: CD 45 D6 21 6F E9 AF 06 03 CD 45 D6 21 81 E9 3E
D560: FE 06 05 CD 45 D6 21 9F E9 3E FC 06 07 CD 45 D6
D570: 21 C9 E9 3E FA 06 09 CD 45 D6 21 6E E9 3E F3 06
D580: 03 CD 45 D6 21 80 E9 3E F2 06 05 CD 45 D6 21 9E
D590: E9 3E F1 06 07 CD 45 D6 21 C8 E9 3E F0 06 09 CD
D5A0: 45 D6 21 6D E9 3E 1A 06 03 CD 4D D6 21 7F E9 3E
D5B0: 23 06 05 CD 4D D6 21 9D E9 3E 2C 06 07 CD 4D D6
D5C0: 21 C7 E9 3E 35 06 09 CD 4D D6 3E 18 32 6A E9 18
D5D0: 06 00 00 00 00 00 00 21 B0 F9 22 68 E9 3E 1F 32
D5E0: 67 E9 3E 0E 32 66 E9 AF 32 65 E9 32 62 E9 32 50
D5F0: E9 32 31 E9 32 37 E9 32 3E E9 32 43 E9 32 48 E9
D600: 32 52 E9 32 57 E9 32 5C E9 21 93 F9 22 53 E9 21
D610: 82 FA 22 58 E9 21 EB FB 22 5D E9 21 0E 02 22 55
D620: E9 21 10 01 22 5A E9 21 13 02 22 5F E9 3E 60 32
D630: 42 E9 32 4C E9 3E 44 32 47 E9 C3 50 E0 00 00 00
D640: 00 00 00 06 18 11 06 00 77 19 10 FC C9 11 06 00
D650: 77 19 C6 07 10 FA C9 00 00 00 06 19 11 78 00 77
D660: 19 10 FC C9 00 00 00 21 A2 F8 06 42 36 24 23 10
D670: FB 21 01 F3 3E 97 CD 5A D6 21 44 F3 3E 88 C3 5A
D680: D6 00 00 00 00 36 CC 23 36 CC 23 36 EF 23 36 CC
D690: 23 36 CC 23 36 4C C9 00 00 00 00 E5 C5 06 06 36
D6A0: 00 23 10 FB C1 E1 C9 00 00 00 21 E8 F8 CD 85 D6
D6B0: 21 E8 FD 06 09 CD 9B D6 11 88 FF 19 E5 CD 85 D6
D6C0: E1 16 80 CD CE D6 10 ED C9 00 00 00 16 07 1E 00
D6D0: 1B 7A B3 20 FB C9 00 00 00 C5 3A 29 E9 47 87 87
D6E0: 80 3C 32 29 E9 C1 C9 00 00 00 01 10 27 CD 06 D7
D6F0: 01 E8 03 CD 06 D7 01 64 00 CD 06 D7 0E 0A CD 06
D700: D7 7D C6 30 12 C9 C5 37 3F 3E 30 ED 42 38 03 3C
D710: 18 F9 C1 09 12 13 C9 00 00 00 21 FD E9 34 3A FF
D720: E9 FE 01 C0 DB 09 FE FF C0 AF 32 FF E9 C9 00 00
D730: 00 3A FD E9 3C E6 0F C0 DD 21 6C E9 3A 6B E9 FE
D740: 00 20 04 3E 01 18 04 FE 05 C0 AF 32 FE E9 06 18
D750: DD 7E 00 FE 01 28 09 11 06 00 DD 19 10 F2 C9 00
D760: DD 34 03 DD 7E 03 4F FE 0A 20 0A 3E 02 DD 77 03
D770: 11 B8 FC 18 03 11 78 00 DD 6E 01 DD 66 02 E5 19
D780: DD 75 01 DD 74 02 E1 A7 79 FE 0C 30 CA 0E 04 7E
D790: 36 00 E5 19 77 E1 23 0D 20 F5 C3 57 D7 3A FD E9
D7A0: E6 05 C0 DD 21 6C E9 06 18 DD 7E 00 FE 01 28 0B
D7B0: 11 06 00 DD 19 10 F2 C9 00 00 00 DD 6E 01 DD 66
D7C0: 02 A7 DD 7E 03 FE 0C D2 C0 D8 E5 36 00 23 23 23
D7D0: 36 00 E1 DD 7E 05 CB 67 E5 20 1E FE 01 20 05 CD
D7E0: 72 D8 18 0C FE 02 20 05 CD 90 D8 18 03 CD B0 D8
D7F0: DD CB 05 E6 18 1C 00 00 00 FE 11 20 05 CD 63 D8
D800: 18 0C FE 12 20 05 CD 81 D8 18 03 CD A0 D8 DD CB
D810: 05 A6 E1 3A FD E9 E6 0F C2 B0 D7 3A FE E9 FE 00
D820: 28 1C DD 34 04 23 DD 75 01 DD 74 02 0E 04 23 23
D830: 23 2B 7E 23 77 2B 0D 20 F8 18 17 00 00 00 DD 35
D840: 04 2B DD 75 01 DD 74 02 0E 04 23 7E 2B 77 23 0D
D850: 20 F8 36 00 DD 7E 04 32 6B E9 C3 B0 D7 00 00 00
D860: 00 00 00 36 C8 23 36 B6 23 36 6B 23 36 8C C9 00
D870: 00 00 36 22 23 36 B6 23 36 6B 23 36 22 C9 00 00
D880: 00 36 C8 23 36 E5 23 36 5E 23 36 8C C9 00 00 00
D890: 36 46 23 36 E5 23 36 5E 23 36 64 C9 00 00 00 00
D8A0: 36 84 23 36 6C 23 36 C6 23 36 48 C9 00 00 00 00
D8B0: 36 A8 23 36 6C 23 36 C6 23 36 8A C9 00 00 00 00
D8C0: DD CB 05 66 20 06 DD CB 05 E6 18 04 DD CB 05 A6
D8D0: 3A FD E9 E6 0F C2 B0 D7 3A FE E9 FE 00 28 06 DD
D8E0: 34 04 23 18 04 DD 35 04 2B DD 75 01 DD 74 02 C3
D8F0: 54 D8 00 00 00 00 3A FD E9 E6 03 C0 DB 00 FE EF
D900: CA 21 D9 FE BF CA 44 D9 3A FD E9 E6 07 C0 DB 01
D910: FE FE CA 91 D9 DB 00 FE FB CA A7 D9 C9 00 00 00
D920: 00 3A 67 E9 3D FE 01 C8 2A 68 E9 2B 7E FE 00 C2
D930: 20 E0 22 68 E9 CD 85 D6 23 36 00 21 67 E9 35 C9
D940: 00 00 00 00 3A 67 E9 3C FE 3D C8 2A 68 E9 11 06
D950: 00 19 7E FE 00 C2 20 E0 2A 68 E9 36 00 23 22 68
D960: E9 CD 85 D6 21 67 E9 34 C9 00 00 00 00 2A 68 E9
D970: E5 19 22 68 E9 E1 06 06 7E 4F 36 00 E5 19 7E FE
D980: 00 28 05 3E 01 32 65 E9 71 E1 23 10 EB C9 00 00
D990: 00 3A 66 E9 3D FE 0D C8 11 88 FF CD 6D D9 21 66
D9A0: E9 35 18 14 00 00 00 3A 66 E9 3C FE 19 C8 11 78
D9B0: 00 CD 6D D9 21 66 E9 34 3A 65 E9 FE 01 C0 C3 20
D9C0: E0 00 00 00 00 3A 64 E9 FE 03 20 04 3E 02 18 06
D9D0: FE 09 20 05 3E 03 32 63 E9 CD 28 DA 3A 63 E9 FE
D9E0: 02 28 04 FE 03 20 19 CD 0D DA CD 28 DA CD 0D DA
D9F0: 3A 63 E9 FE 03 20 09 CD 12 DA CD 28 DA CD 12 DA
DA00: 3A 62 E9 FE 00 C8 C3 20 E0 00 00 00 00 21 57 E9
DA10: 18 03 21 5C E9 11 52 E9 06 05 4E 1A 77 79 12 23
DA20: 13 10 F7 C9 00 00 00 00 3A FD E9 E6 0F C0 3A 52
DA30: E9 FE 00 20 0B CD D9 D6 E6 03 C0 3E 01 32 52 E9
DA40: 2A 53 E9 E5 36 00 23 36 00 E1 A7 3A 56 E9 FE 40
DA50: 38 18 AF 32 52 E9 21 DA FC 22 53 E9 3E 15 32 55
DA60: E9 3E 01 32 56 E9 C9 00 00 00 CD D9 D6 E6 07 CA
DA70: 8B DA CD D9 D6 E6 05 CA B3 DA CD D9 D6 E6 01 CA
DA80: CA DA 36 69 23 36 96 C9 00 00 00 3A 56 E9 3C 32
DA90: 56 E9 23 22 53 E9 23 7E FE CC 28 08 FE EF 28 04
DAA0: FE 4C 20 06 3E 01 32 62 E9 C9 36 96 2B 36 69 C9
DAB0: 00 00 00 3A 55 E9 3D FE 0C CA CA DA 11 88 FF 32
DAC0: 55 E9 19 C3 8B DA 00 00 00 00 3A 55 E9 3C FE 19
DAD0: CA B3 DA 11 78 00 C3 BF DA 00 00 00 00 3A 50 E9
DAE0: FE 00 C0 3A FF E9 FE 00 C0 DB 09 FE BF C0 3E 01
DAF0: 32 FF E9 2A 68 E9 23 23 22 4F E9 C9 00 00 00 00
DB00: 00 00 3A FD E9 E6 00 C0 2A 4F E9 7C FE 00 C8 7E
DB10: FE 0E 20 02 36 00 01 88 FF 09 7C FE F2 20 08 AF
DB20: 32 50 E9 C9 00 00 00 22 4F E9 7E FE 96 20 07 36
DB30: 00 2B 36 00 18 0B FE 69 C2 83 DB 36 00 23 36 00
DB40: 2B 11 52 E9 EB 06 03 7E FE 00 20 10 23 23 23 23
DB50: 23 10 F4 AF 32 50 E9 EB C9 00 00 00 23 7B B8 28
DB60: 03 C3 4D DB 23 7A BE 28 03 C3 4E DB 2B 2B 36 00
DB70: 23 36 0B 23 36 FA 23 36 0F 23 36 02 C3 50 DB 00
DB80: 00 00 00 FE 24 C8 A7 7C FE F8 D2 6A DC 7E FE B6
DB90: 28 05 FE 6B 20 08 2B 2B 01 1E 00 C3 BC DB FE E5
DBA0: 28 05 FE 5E 20 08 2B 2B 01 32 00 C3 BC DB FE 6C
DBB0: 28 06 FE C6 C2 BD DC 2B 2B 01 46 00 3A 6A E9 3D
DBC0: 32 6A E9 E5 2A 2F E9 09 22 2F E9 11 28 F5 CD EA
DBD0: D6 E1 11 6C E9 EB 06 18 7E FE 00 20 0B 23 23 23
DBE0: 23 23 23 10 F3 18 15 00 23 7B BE 28 03 C3 DE DB
DBF0: 23 7A BE 28 03 C3 DF DB 2B 2B 36 00 EB CD B0 DF
DC00: 00 16 04 36 80 23 36 25 23 36 A4 23 36 01 D5 CD
DC10: CC D6 D1 36 85 2B 36 2C 2B 36 34 2B 36 A1 D5 CD
DC20: CC D6 D1 15 20 DD 3E 00 D3 40 36 00 23 36 00 23
DC30: 36 00 23 36 00 CD 3D DC AF 32 50 E9 C9 2A 2F E9
DC40: 3A 2C E9 FE 00 C0 7C FE 0F 20 07 A7 7D FE A0 D8
DC50: 18 04 A7 FE 0F D8 3A 3D E9 3C 32 3D E9 C6 30 21
DC60: DA F9 77 3E 01 32 2C E9 C9 00 7E FE 22 20 11 2B
DC70: 7E FE 6B 28 03 23 18 02 2B 2B 01 32 00 C3 BC DB
DC80: FE B6 28 F5 FE 6B 28 F0 FE 46 28 0F FE E5 28 0A
DC90: FE 5E 28 05 FE 64 20 0C 2B 2B 2B 01 64 00 C3 BC
DCA0: DB 00 00 00 FE A8 28 0F FE 6C 28 0A FE C6 28 05
DCB0: FE 8A 20 09 2B 2B 2B 01 96 00 C3 BC DB FE 00 20
DCC0: 03 36 0E C9 AF 32 50 E9 C9 00 00 00 00 00 00 00
DCD0: 3A 3E E9 FE 00 20 03 CD 10 DD 3A 43 E9 FE 00 20
DCE0: 09 CD FD DC CD 10 DD CD FD DC 3A 48 E9 FE 00 C0
DCF0: CD 02 DD CD 10 DD C3 02 DD 00 00 00 00 21 43 E9
DD00: 18 03 21 48 E9 11 3E E9 C3 18 DA 00 00 00 00 00
DD10: CD D9 D6 E6 24 C0 DD 21 6C E9 06 18 DD 7E 00 FE
DD20: 01 28 0B 11 06 00 DD 19 10 F2 C9 00 00 00 A7 DD
DD30: 7E 03 FE 0C 30 ED CD D9 D6 E6 7F 20 E6 DD 6E 01
DD40: DD 66 02 23 11 3E E9 EB 36 01 23 73 23 72 DD 7E
DD50: 03 23 77 C9 00 00 00 00 3A FD E9 E6 03 C0 3A 3E
DD60: E9 FE 01 20 03 CD 8B DD 3A 43 E9 FE 01 20 09 CD
DD70: FD DC CD 8B DD CD FD DC 3A 48 E9 FE 01 C0 CD 02
DD80: DD CD 8B DD C3 02 DD 00 00 00 00 2A 3F E9 7E FE
DD90: 60 20 04 3E 44 18 06 FE 44 20 07 3E 60 32 42 E9
DDA0: 36 00 01 78 00 09 3A 41 E9 3C FE 19 20 05 AF 32
DDB0: 3E E9 C9 37 3F FE 0D 30 12 32 41 E9 22 3F E9 7E
DDC0: FE 0E 28 03 FE 00 C0 36 60 C9 00 3A FD E9 E6 07
DDD0: C2 DD DD 3A 41 E9 3C 32 41 E9 22 3F E9 7E FE 4C
DDE0: 20 08 2B 2B 2B 2B 2B C3 20 E0 FE EF 20 04 2B 2B
DDF0: 18 F5 FE CC 20 10 23 7E FE EF 28 F2 FE 4C 28 E2
DE00: 23 18 F4 00 00 00 FE 0E 28 03 FE 00 C0 3A 42 E9
DE10: 77 C9 00 00 00 00 2A 32 E9 36 00 23 36 00 23 36
DE20: 00 23 36 00 21 F2 F3 3A 6B E9 5F 16 00 06 09 19
DE30: 7E FE 00 28 0D 23 23 23 23 23 23 23 10 F2 C3 24
```

マシン語プログラム　ダンプ・リスト

```
DE40: DE 00 CD D9 D6 E6 03 20 EC 06 18 DD 21 6C E9 DD
DE50: 7E 00 FE 00 28 0B 11 06 00 DD 19 10 F2 C9 00 00
DE60: 00 DD 36 00 01 DD 75 01 DD 74 02 DD 36 03 02 3A
DE70: 6B E9 DD 77 04 3A 35 E9 DD 77 05 C9 00 00 00 00
DE80: 3A FD E9 E6 03 C0 3A 31 E9 FE 00 20 05 CD C2 DE
DE90: 18 03 CD 3A DF CD AD DE 3A 31 E9 FE 00 20 05 CD
DEA0: C2 DE 18 03 CD 3A DF CD AD DE C9 00 00 21 37 E9
DEB0: 11 31 E9 06 06 4E 1A 77 79 12 23 13 10 F7 C9 00
DEC0: 00 00 DD 21 6C E9 06 18 DD 7E 00 FE 01 28 0B 11
DED0: 06 00 DD 19 10 F2 C9 00 00 00 DD 7E 03 FE 09 20
DEE0: EE CD D9 D6 E6 03 20 E7 DD 6E 01 DD 66 02 DD 36
DEF0: 00 00 E5 36 00 23 36 00 23 36 00 23 36 00 E1 11
DF00: E0 01 19 EB 21 31 E9 36 01 23 73 23 72 23 36 0D
DF10: DD 7E 05 23 77 23 A7 3A 32 E9 FE 3B 30 04 3E 7A
DF20: 18 02 3E 76 77 EB 3A 35 E9 E6 02 CA 72 D8 3A 35
DF30: E9 E6 01 CA 90 D8 C3 B0 D8 00 2A 32 E9 3A 36 E9
DF40: 5F 16 00 06 04 E5 19 3A 34 E9 3C FE 19 20 0B E1
DF50: AF 32 31 E9 C3 16 DE 00 00 00 32 34 E9 22 32 E9
DF60: E1 4E 36 00 E5 19 7E FE 4C 28 08 FE EF 28 04 FE
DF70: CC 20 16 22 2A E9 E1 36 00 23 10 FB AF 32 31 E9
DF80: 2A 2A E9 C3 DD DD 00 00 00 FE 0E 20 17 22 2A E9
DF90: E1 36 00 23 10 FB AF 32 31 E9 2A 2A E9 79 C3 6B
DFA0: DC 00 00 00 71 E1 23 05 C2 61 DF C9 00 00 00 00
DFB0: E5 55 3A 32 E9 BA 20 0D 5C 3A 33 E9 BB 20 17 AF
DFC0: 32 31 E9 18 11 3A 38 E9 BA 20 0B 5C 3A 39 E9 BB
```

```
DFD0: 20 04 AF 32 37 E9 E1 3E 20 D3 40 C9 00 00 00 00
DFE0: 00 00 00 CD 21 D5 CD 67 D6 CD AA D6 CD 1A D7 CD
DFF0: 31 D7 CD 9D D7 CD F6 D8 CD C5 D9 CD DD DA CD 02
E000: DB CD D0 DC CD 58 DD CD 80 DE 3A 62 E9 FE 01 C8
E010: 3A 6A E9 FE 00 C8 16 09 CD CE D6 C3 EC DF 00 00
E020: 3E 01 32 62 E9 C9 00 00 00 00 E5 5E 23 66 6B CD
E030: 34 E0 E1 C9 3E 20 D3 40 CD 46 E0 3E 00 D3 40 CD
E040: 46 E0 25 20 EF C9 E5 2D 20 FD E1 C9 00 00 00 00
E050: 3A 64 E9 FE 01 20 08 21 00 00 3E 00 C3 B1 E0 FE
E060: 03 20 08 21 78 00 3E 01 C3 B1 E0 FE 05 20 08 21
E070: F0 00 3E 02 C3 B1 E0 FE 08 20 08 21 68 01 3E 03
E080: C3 B1 E0 FE 0B 20 08 21 E0 01 3E 04 C3 B1 E0 FE
E090: 0F 20 08 21 58 02 3E 05 C3 B1 E0 FE 13 20 08 21
E0A0: D0 02 3E 06 C3 B1 E0 FE 18 C2 B7 E0 21 48 03 3E
E0B0: 07 22 27 E9 32 4E E9 DD 21 6C E9 06 18 DD 6E 01
E0C0: DD 66 02 ED 5B 27 E9 19 DD 75 01 DD 74 02 3A 4E
E0D0: E9 DD 86 03 DD 77 03 11 06 00 DD 19 10 DF C9 00
E0E0: 00 00 00 11 E8 03 2A 2F E9 19 22 2F E9 C9 00 00
E0F0: 00 3E 98 32 55 F3 32 CD F3 32 45 F4 32 BD F4 3E
E100: D8 32 35 F5 32 AD F5 32 25 F6 32 9D F6 3E 58 32
E110: 15 F7 32 8D F7 32 05 F8 32 7D F8 32 6D F9 3E 38
E120: 32 F5 F8 3E B8 32 F9 F8 32 E5 F9 32 5D FA 32 D5
E130: FA 32 4D FB 32 C5 FB 32 3D FC 32 B5 FC 32 2D FD
E140: 32 A5 FD 32 1D FE 3E 78 32 95 FE C9 00 00 00 00
```

■チェックサム・プログラムの使い方

ＲＵＮさせると，START＝ と表示されるので，開始番地を16進数で入力してください．次にＥＮＤ＝と表示されてから，終了番地を16進数で入力します．最後にBYTE＝と表示されるので，計算するバイト数を10進数で入力します．

このCHECK SUMは64バイトでやっています．ＴＶとプリンタの両方へ同時に出力するようにしていますから，プリンタのある人はその方が見やすいでしょう．TVのときは，画面をはみ出すとスクロールされますから，適当なところで ESC キーを使って止めてチェックしてください．

現在のプログラムでは，ＣＡ0·1番地以後にマシン語プログラムがある場合しか使えません．もっともマシン語はたいてい後半に置くので，さしつかえないでしょう．

チェックサム　プログラム・リスト

```
90 'PC-8001 CHECK SUM   "CS"  VER 1.0
100 CLEAR300,&HCA00
110 WIDTH40,25:CONSOLE0,25,0,0
120 PRINTCHR$(12):PRINT"PC-8001 CHECK SUM"
130 INPUT"START=";A$:S=VAL("&H"+A$):IF S<0THEN S=S+65536!
140 INPUT"END  =";B$:E=VAL("&H"+B$):IF E<0THEN E=E+65536!
150 INPUT"BYTE =";T
160 X=0:R=S+T:IF R>=E THEN R=E+1
170 FOR I=S TO R-1
180 Y=PEEK(I)
190 X=X+Y:NEXT
200 K=S:GOSUB260:S$=K$
210 S=S+T:K=R-1:GOSUB260:E$=K$
220 K=X:GOSUB260:X$=K$
230 PRINTS$;" - ";E$;" = ";X$
240 LPRINTS$;" - ";E$;" = ";X$
250 IF S>=E THEN END ELSE GOTO 160
260 U=INT(K/4096):K=K-4096*U:K$=HEX$(U)
270 U=INT(K/256):K=K-256*U:K$=K$+HEX$(U)
280 U=INT(K/16):K=K-16*U:K$=K$+HEX$(U)+HEX$(K)
290 RETURN
```



マシン語プログラムのチェックサム

```
START=D500
END  =E14C
BYTE = 64
D500 - D53F = 182E
D540 - D57F = 1D28
D580 - D5BF = 1C16
D5C0 - D5FF = 18F4
D600 - D63F = 19D1
D640 - D67F = 139B
D680 - D6BF = 1ABC
D6C0 - D6FF = 1777
D700 - D73F = 1F7C
D740 - D77F = 160D
D780 - D7BF = 1B2D
D7C0 - D7FF = 1AF7
```

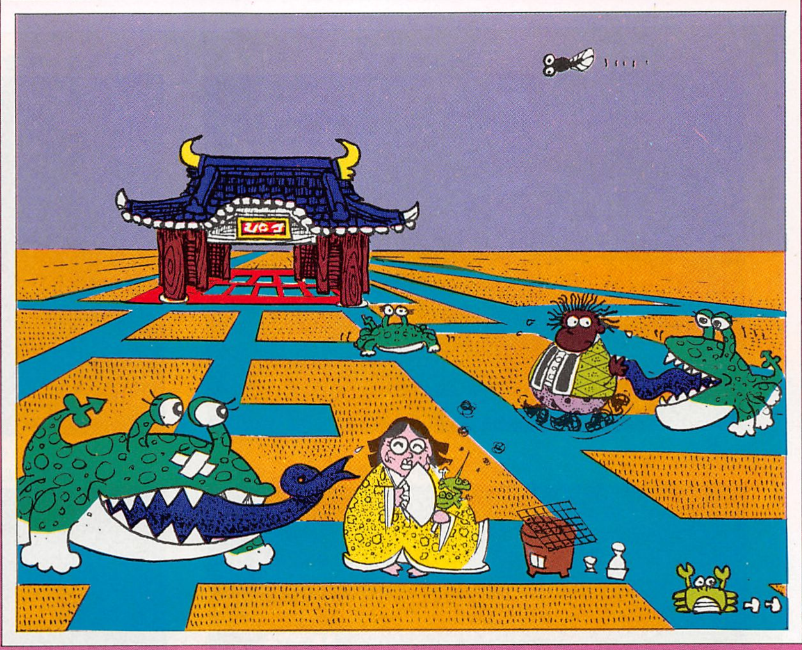
```
D800 - D83F = 1A01
D840 - D87F = 12C5
D880 - D8BF = 110E
D8C0 - D8FF = 1DA1
D900 - D93F = 22FD
D940 - D97F = 18E5
D980 - D9BF = 1A48
D9C0 - D9FF = 19B8
DA00 - DA3F = 18FB
DA40 - DA7F = 2099
DA80 - DABF = 1AA1
DAC0 - DAFF = 1C35
DB00 - DB3F = 18B7
DB40 - DB7F = 1570
DB80 - DBBF = 1C6F
```

```
DBC0 - DBFF = 1B7C
DC00 - DC3F = 175D
DC40 - DC7F = 1ACF
DC80 - DCBF = 1AE4
DCC0 - DCFF = 1B40
DD00 - DD3F = 1AD5
DD40 - DD7F = 1E37
DD80 - DDBF = 1756
DDC0 - DDFF = 1E17
DE00 - DE3F = 120F
DE40 - DE7F = 1856
DE80 - DEBF = 1CF4
DEC0 - DEFF = 1671
DF00 - DF3F = 1B04
DF40 - DF7F = 195A
```

```
DF80 - DFBF = 1940
DFC0 - DFFF = 1FAF
E000 - E03F = 2107
E040 - E07F = 177B
E080 - E0BF = 1AA0
E0C0 - E0FF = 1977
E100 - E13F = 2270
E140 - E14C = 0665
```

PC-8001 (BASIC＋マシン語)

TOMグループ
塘田一夫
岡部和義
松江　亨



パックマン

パックマンはゲームセンターに登場して，だいぶ月日が経ち人気が下がってきていますが，ここにＰＣ版のパックマンを発表します．

最初バグが大量発生してしまい，マシン語の最後の方はバグ取りルーチンだらけになっていますが，割と本物に近くできたので結構楽しめます．

効果音もＰＣとしてはいい音を出していると思って満足しています．

遊び方

ルールは分かっていると思いますが，パックマン(■)が4匹のモンスター(♣)につかまらないようにしながら，全部のドットを食べてしまうゲームです．

ドットの中にはパワーエサという物があって，パックマンがそれを食べるとモンスター達は♥型になり，青ざめていじけてしまいます．パックマンはそのときにドットを食べていてもよいのですが，モンスター達を食べて反撃することができます．

そのときの点数は1匹食べるごとに200，400，800，1600とはね上がっていきます．モンスターは食べられると，少したってまた中央から出てきて，いじけていた奴も点滅して元に戻ります．

全部のドットを食べ尽すと次の面に移ります．また，画面の中央に変な文字が出てきますが，これはフルーツで，BONUS得点です．

まず，BASICとマシン語のプログラムを入れてＲＵＮさせてください．モンスターのキャラクタとニックネームが出てきます．

アカベーはパックマンを追いかけてきて，ピンキーは待ち伏せしてきます．アオスケは追いかけるのも気分次第で，グズタはただ歩き回っているだけです．

使用するキーはテンキーの 2 ， 4 ， 6 ， 8 です．パックマンは壁にぶつからない限り，動き続けるので，キーを押さなくても前進し続けます．また，キーを押した方向が壁だったら，パックマンはその方向を向かずにやはり前進し続けます．

モンスターやフルーツを食べたときの点数は画面右下にずっと出ています．10,000点に達するとパックマンが1匹あるいは1人増えます．さあ，心の準備ができたら RETURN キーを押してください．画面が消え（OUT81，0），タイトル・ミュージックが鳴ったらスタートです．なお，655,350点を超えたときはモンスターがギブアップして終わります．

プログラムの説明

非効率的で簡単なプログラムですが，サブルーチンとワーク・エリアおよびデータ・エリアを書いておきます（表

写真１　ゲームの説明

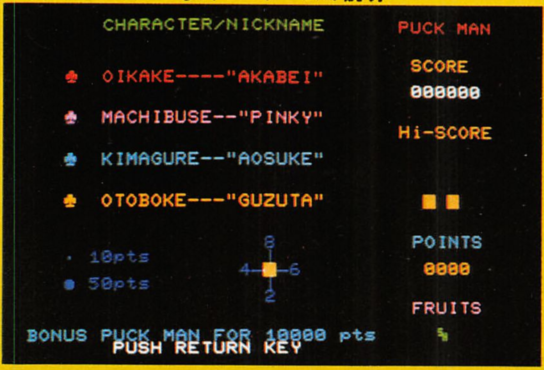


写真２　アカベーとピンキーが現われる

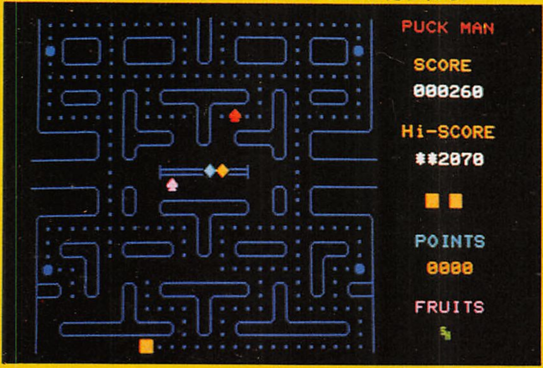


写真３　パックマンがモンスターにつかまり，しぼんでいくところ

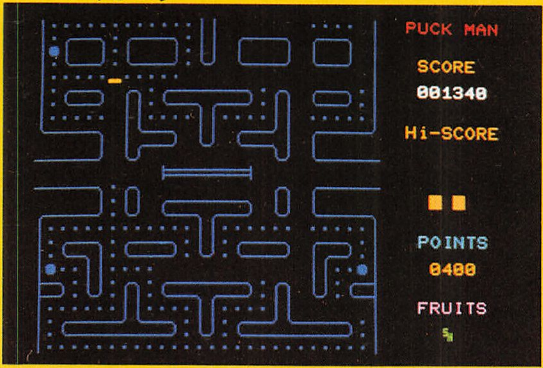


写真４　つかまった後は，その位置からゲームが始まる

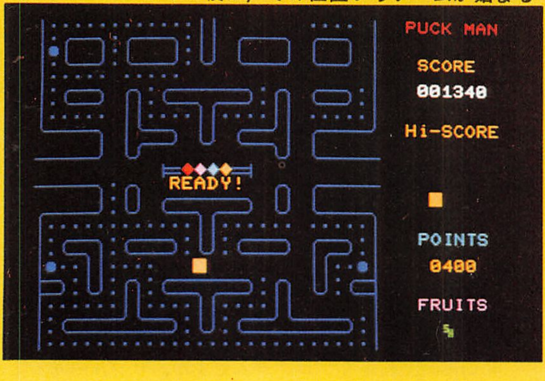


写真５　栄養エサを食べて，パックマンの逆襲！

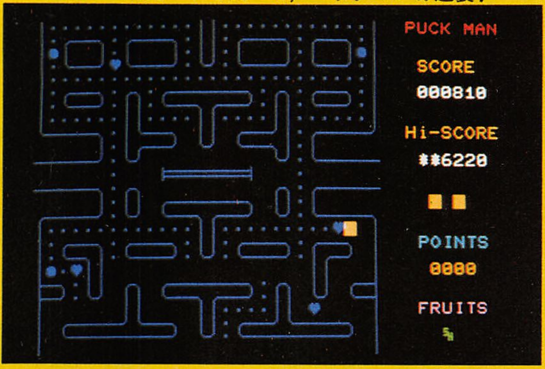


写真６　画面の中央には，フルーツが…

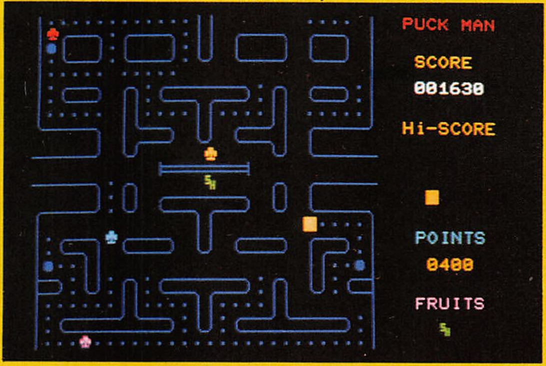


写真７　ここまでくれば，１面は楽勝（？）

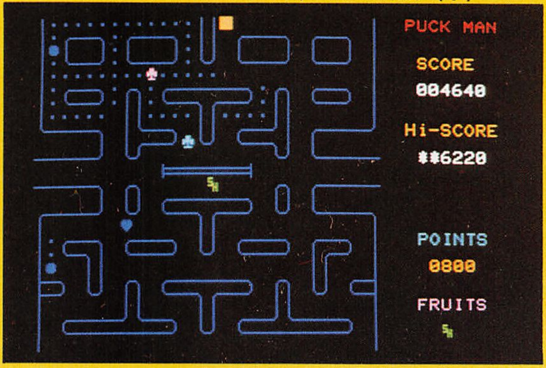


写真８　ゲーム・オーバー

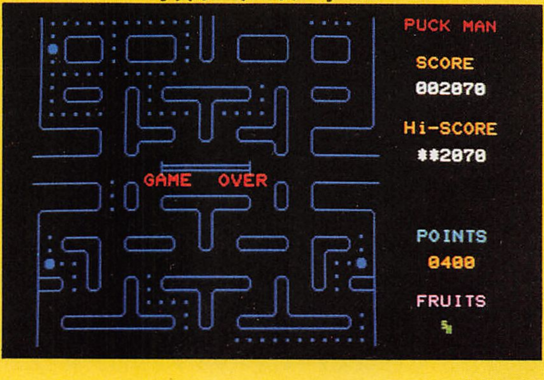


図1　ジェネラル・フローチャート

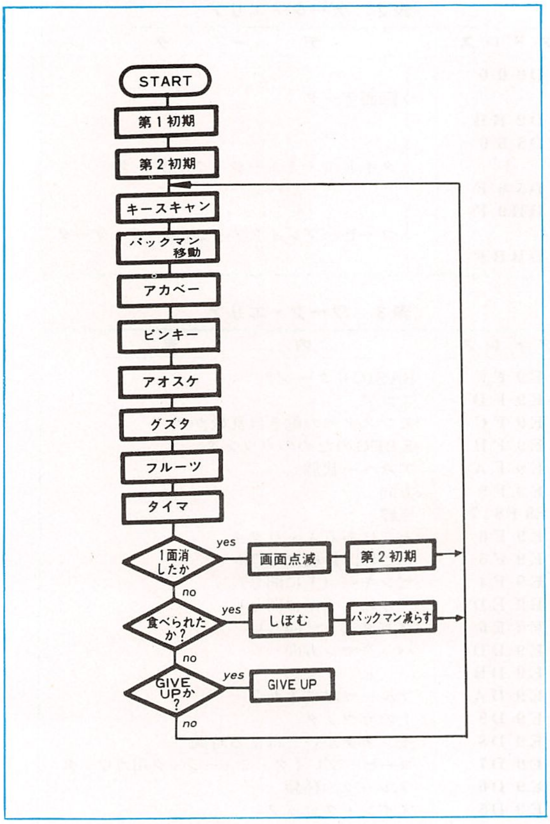
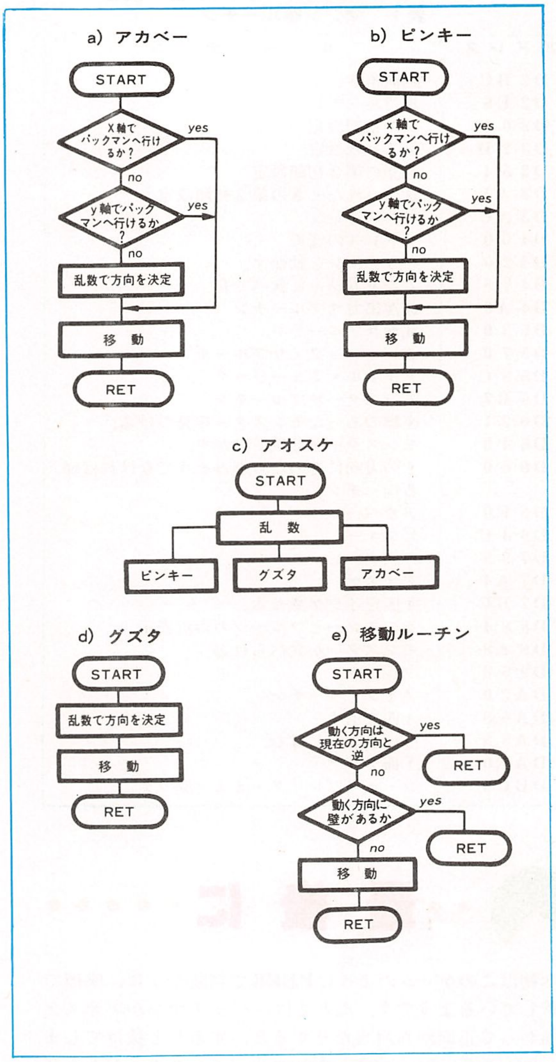


図2　各モンスターのフローチャート



1～3)．マシン語はD000H～DC13Hです．

フルーツは友人からドットを70個か140個消したときに出てくると聞いたものですから，そのようにしました．

バグは一応全部取ったつもりですが，もしかしたら1面消しても終わらないかも知れません（バグを取る前がそうだったので）．そのときはあきらめてください．

今のところこのようなバグは出ていません．また得点の最後の0はニセ物で，6,500点の場合，ワーク・エリアには650点という数値が入っています．

タイトル・ミュージックやコーヒーブレイク・ミュージックはI/O'80年11月号に載っていたミュージック・サブルーチンを拝借させていただきました．

BASICのプログラムに**CLEAR**文は入っていませんが必要であれば最初の部分に**CLEAR300, & HCFFF**を入れてください．

380行の**PRINTUSING**“＊＊＃＃＃0”；**HS**の2つのアスタリスク(＊)は数字の桁が空いているところを＊で埋めることを意味します．いらなければ＊を＃に換えてください

パックマンはPACK MANと書く人が多いのですが，このリストのようにPUCK MANと書くのが正しい綴です．手抜きとして画面のドットに色を付けていません．

OUT81, 0は画面がパッと出るようにと音が良くなるように入れてあるのですが，異常動作をする場合は取った方がいいかも知れません．取るときは0DBC0Hからの3バイトも00にしてください．

プログラムのジェネラル・フローチャートを図1に，各モンスターのフローチャートを図2に示します．また，プログラムの変更点を表4に示します．

高得点へのコツ

❶モンスターがいじけているときには初心者の場合ドットを食べ続けて食べられないよう心掛ける，プロの場合はなるべく食べる．

❷モンスターを食べようと思ったら，まずすべてのモンスターを引きつけておいてパワーエサを食べる．

❸モンスターが散らばっているときにフルーツが出てきたら，両側からはさまれるので引きつけておいたまま食べにいく．

❹各モンスターの性質を理解する．

❺ワープ・トンネルをとおるモンスターは警戒して速度が遅くなるので，引き離そうと思ったらそれを利用する．

❻❺を乱用しない．逃げ場がないので，はさまれる危険がある．

❼時々あるモンスターの向きの反転には注意する．

❽パワーエサを食べてもいじけなくなったら❺を使う．

表１　マシン語ルーチン

アドレス	ルーチン
D２BC	画面書き
D２E６	単音ルーチン
D３０４	第１初期設定
D３２D	第２初期設定
D３５４	最初の第２初期設定
D３A７	食べられたときの第２初期設定
D３AC	キースキャン
D４００	アドレスの変更
D４２０	パックマンを動かす．
D４７８	ムシャムシャと食べる音
D４A０	10進出力サブルーチン
D５３９	乱数サブルーチン
D５７０	ミュージック・サブルーチン
D５８C	タイトル・ミュージック
D５B７	モンスターサブルーチン
D６２１	座標のあったモンスターを見つける．
D６４３	モンスターを乱数で動かす．
D６６０	その方向に進めたら進みそうでなければ帰るルーチン
D６F０	アカベー
D６４C	ピンキー
D７９８	グズタ
D７A４	アオスケ
D７BC	４匹をイジケさせる．
D８８４	モンスターとフルーツの得点表示
D８A８	モンスターが食べられる．
D９５９	フルーツ
DA２０	メイン・ルーチン
DA６８	４匹を消す．
DA８８	パックマンしぼむ．
DAE０	１面クリア
DBC０	コーヒーブレイク・ミュージック

表２　データ・エリア

アドレス	データ
D０００ 〜 D２BB	画面データ
D５５０ 〜 D５６F	タイトル・ミュージック・データ
DB９F 〜 DBBF	コーヒーブレイク・ミュージック・データ

表３　ワーク・エリア

アドレス	内容
E９FF	BASICリターン？
E９FD	スコア
E９FC	モンスターの動きは乱数か？
E９FB	E９FCのためのカウンタ
E９FA	アカベー状態
E９F９	方向
E９F８,７	座標
E９F６	そこにあるキャラクタ
E９F５	いじけ，待機のカウンタ
E９F４	ピンキー（上に同じ）
E９ED	アオスケ（上に同じ）
E９E６	グズタ（上に同じ）
E９DD	パックマン方向
E９DB	〃　座標
E９DA	フルーツ出てるか？
E９D９	上のカウンタ
E９D８	モンスタがいじけてる時間
E９D７	コーヒーブレイク・ミュージック用カウンタ
E９D６	フルーツの種類
E９D５	メイン・クロック
E９D４	パックマンの数
E９D３	数を増やしたか？
E９D１	HLの一時スタック
E９CF	ムシャムシャと食べる音の上がり下がり
E９CE	ドットの残り数
E９CC	乱数
E９CB	色スタック

表４　変更点

アドレス	現在のデータ	データの内容
D３２７	５０	１番最初のモンスターがいじけている時間
DB５３	５０	同じくコーヒーブレイク・ミュージックが鳴ってからのいじけている時間
D９７E	４０	フルーツが出ている時間

注）パックマンの数…，書こうと思いましたがやめました．パックマンの数はこれで充分です．どうしてもという人はプログラムを解読してみてください．全体の速度は１面消すごとに速くなってくるので変更は不可能です．

最後に

本物はこのゲームのようにPEEKで判断せずに，座標で判断しているようです．たとえば，パックマンがアオスケに向かって正面から対当たりすると，するりと抜けてしまうときがあるというバグがあります（モンスターだから意識的に作ったのかも知れない）．

このプログラムはそんなことはありませんが意識的に作ったものとして，

1）モンスターを食べるときに，そこにあったドットの点数は加算されない．

というのがあります．後者のはパワーエサとモンスターが重なっているときに食べなければどうもないでしょう．

なお，私達の最高得点は岡部和義君の84,670点という記録です．書き忘れていましたが，パターンはゲームを面白くするために作っていません．

次はアイデアともにオリジナルの作品を作ろうと思います．

◘参考文献

1）芸夢狂人："ALIEN FALL"，I/O，'80年11月号

2）長谷正博："PC-8001モニタ・サブルーチンの利用法"，I/O，'80年11月号

パックマン　BASICプログラム

```
10 '********************************
20 '*      TOM ORIGINAL SOFT       *
30 '*     PUCK MAN        G-101    *
40 '********************************
50 CONSOLE0,25,0,1:COLOR7,0,0:WIDTH40,25:PRINTCHR$(12)
60 COLOR2:LOCATE30,0:PRINT"PUCK MAN
70 COLOR5:LOCATE31,16:PRINT"POINTS
80 COLOR6:LOCATE31,3:PRINT"SCORE
90 LOCATE30,8:PRINT"Hi-SCORE
```

パックマン BASICプログラム

```
100 COLOR3:LOCATE31,21:PRINT"FRUITS
110 GOSUB420
120 DEFUSR=&HD304:A=USR(0)
130 LOCATE6,0:PRINT"CHARACTER/NICKNAME"
140 COLOR2:LOCATE3,4:PRINT"♣  OIKAKE----"CHR$(34)"AKABEI"CHR$(34)
150 COLOR3:LOCATE3,7:PRINT"♣  MACHIBUSE--"CHR$(34)"PINKY"CHR$(34)
160 COLOR5:LOCATE3,10:PRINT"♣  KIMAGURE--"CHR$(34)"AOSUKE"CHR$(34)
170 COLOR6:LOCATE3,13:PRINT"♣  OTOBOKE---"CHR$(34)"GUZUTA"CHR$(34)
180 COLOR1:LOCATE3,17:PRINT"· 10pts
190 LOCATE3,19:PRINT"● 50pts
200 LOCATE18,16:PRINT8
210 LOCATE19,17:PRINT"|
220 LOCATE17,18:PRINT"4-";:COLOR6:PRINT"■";:COLOR1:PRINT"-6
230 LOCATE19,19:PRINT"|
240 LOCATE18,20:PRINT2
250 COLOR5:LOCATE0,23:PRINT"BONUS PUCK MAN FOR 10000 pts
260 COLOR7:LOCATE7,24:PRINT"PUSH RETURN KEY";:LINE24,1:WAIT1,128,255
270 OUT81,0:LINE(0,0)-(27,24)," ",1,BF:LINE24,0:DEFUSR=&HD597:A=USR(0):GOSUB420
280 LOCATE12,11:COLOR2:PRINT"♦";:COLOR3:PRINT"♦";:COLOR5:PRINT"♦";:COLOR6:PRINT"
♦
290 COLOR6:LOCATE11,12:PRINT"READY!
300 LOCATE13,18:PRINT"■
310 WIDTH40
320 FORI=1TO1500:NEXT:COLOR1:LOCATE11,12:PRINT"      "
330 DEFUSR=&HDA20:A=USR(0)
340 IFPEEK(&HE9CE)=0GOTO350ELSEONPEEK(&HE9FF)-1GOTO360,410
350 POKE-500,PEEK(&HE9D6):OUT81,0:DEFUSR=&HD354:A=USR(0):GOTO280
360 A=PEEK(&HE9D4):IFA=0GOTO370ELSELOCATE30+A*2,13:COLOR6:PRINT" ":DEFUSR=&HD3A7
:A=USR(0):GOTO280
370 COLOR2:LOCATE9,12:PRINT"GAME  OVER":S=PEEK(&HE9FD)+PEEK(&HE9FE)*256:IFS>HSTH
ENHS=S
380 COLOR7:LOCATE31,10:PRINTUSING"**###0";HS
390 FORI=1TO2000:NEXT
400 OUT81,0:LINE(0,0)-(27,24)," ",1,BF:WIDTH40:GOTO120
410 COLOR7:LOCATE10,12:PRINT"GIVE  UP"CHR$(7):HS=65535!:GOTO380
420 COLOR7:LOCATE31,5:PRINT"000000"
430 COLOR6:LOCATE32,18:PRINT"0000"
440 COLOR6:LOCATE32,13:PRINT"■ ■  "
450 COLOR4:LOCATE33,23:PRINTCHR$(1)
460 RETURN
```

パックマン マシン語プログラム

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D000 96 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 96 96 A5 :A23
D010 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 96 96 A5 9C 95 :A19
D020 95 9D A5 9C 95 95 95 9D A5 96 96 A5 9C 95 95 95 :9A0
D030 9D A5 9C 95 95 9D A5 96 96 EC 96 20 20 96 A5 96 :909
D040 20 20 20 96 A5 96 96 A5 96 20 20 20 96 A5 96 20 :653
D050 20 96 EC 96 96 A5 9E 95 95 9F A5 9E 95 95 95 9F :97B
D060 A5 9E 9F A5 9E 95 95 95 9F A5 9E 95 95 9F A5 96 :9CA
D070 96 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 :A41
D080 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 96 96 A5 9C 95 :A19
D090 95 9D A5 9C 9D A5 9C 95 95 95 95 95 95 9D A5 9C :9AD
D0A0 9D A5 9C 95 95 9D A5 96 96 A5 9E 95 95 9F A5 96 :9BD
D0B0 96 A5 9E 95 95 9D 9C 95 95 9F A5 96 96 A5 9E 95 :9AE
D0C0 95 9F A5 96 96 A5 A5 A5 A5 A5 A5 96 96 A5 A5 A5 :9FE
D0D0 A5 96 96 A5 A5 A5 A5 96 96 A5 A5 A5 A5 A5 A5 96 :A05
D0E0 9E 95 95 95 95 9D A5 96 9E 95 95 9D 20 96 96 20 :89B
D0F0 9C 95 95 9F 96 A5 9C 95 95 95 95 9F 20 20 20 20 :7AF

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D100 20 96 A5 96 9C 95 95 9F 20 9E 9F 20 9E 95 95 9D :838
D110 96 A5 96 20 20 20 20 20 95 95 95 95 95 9F A5 9E :73C
D120 9F 20 20 20 20 20 20 20 20 20 20 9E 9F A5 9E 95 :4F4
D130 95 95 95 95 20 20 20 20 20 20 A5 20 20 20 E1 E0 :5DA
D140 E8 E8 E8 E8 E0 E3 20 20 20 A5 20 20 20 20 20 20 :728
D150 95 95 95 95 95 9D A5 9C 9D 20 20 20 20 20 20 20 :644
D160 20 20 20 9C 9D A5 9C 95 95 95 95 95 20 20.20 20 :643
D170 20 96 A5 96 96 20 9C 95 95 95 95 95 95 9D 20 96 :814
D180 96 A5 96 20 20 20 20 20 9C 95 95 95 95 9F A5 9E :743
D190 9F 20 9E 95 95 9D 9C 95 95 9F 20 9E 9F A5 9E 95 :88E
D1A0 95 95 95 9D 96 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 :A09
D1B0 A5 96 96 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 96 :A23
D1C0 96 A5 9C 95 95 9D A5 9C 95 95 95 9D A5 96 96 A5 :9B1
D1D0 9C 95 95 95 9D A5 9C 95 95 9D A5 96 96 A5 9E 95 :9A9
D1E0 9D 96 A5 9E 95 95 95 9F A5 9E 9F A5 9E 95 95 95 :9B8
D1F0 9F A5 96 9C 95 9F A5 96 96 EC A5 A5 96 96 A5 A5 :A27

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D200 A5 A5 A5 A5 A5 20 20 A5 A5 A5 A5 A5 A5 A5 96 96 :928
D210 A5 A5 EC 96 93 95 9D A5 96 96 A5 9C 9D A5 9C 95 :A16
D220 95 95 95 95 95 9D A5 9C 9D A5 96 96 A5 9C 95 92 :99D
D230 93 95 9F A5 9E 9F A5 96 96 A5 9E 95 95 9D 9C 95 :9B5
D240 95 9F A5 96 96 A5 9E 9F A5 9E 95 92 96 A5 A5 A5 :9D6
D250 A5 A5 A5 96 96 A5 A5 A5 A5 96 96 A5 A5 A5 A5 96 :A05
D260 96 A5 A5 A5 A5 A5 A5 96 96 A5 9C 95 95 95 95 9F :9D4
D270 9E 95 95 9D A5 96 96 A5 9C 95 95 9F 9E 95 95 95 :99D
D280 95 9D A5 96 96 A5 9E 95 95 95 95 95 95 95 95 9F :98D
D290 A5 9E 9F A5 9E 95 95 95 95 95 95 95 95 9F A5 96 :9A7
D2A0 96 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 :A41
D2B0 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 96 21 02 F3 11 :8D4
D2C0 00 D0 0E 19 06 1C EB 7E EB 77 23 23 13 10 F7 D5 :619
D2D0 11 40 00 19 D1 0D 20 EC C9 D5 7A E5 CD F8 04 E1 :7FB
D2E0 CD F3 03 D1 73 C9 E5 0E 20 53 42 79 D3 40 10 FE :812
D2F0 42 AF D3 40 10 FE 2B 7C B5 C2 EA D2 E1 C9 00 00 :896

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D300 00 00 00 00 21 00 00 22 FD E9 21 D4 E9 3E 03 77 :4BF
D310 AF 21 D3 E9 77 21 D6 E9 3E 01 77 21 D0 E9 3E 01 :7B2
D320 77 21 D7 E9 AF 77 3E 50 21 D8 E9 77 C9 AF 21 FF :8FD
D330 E9 77 21 DA E9 77 2B 77 3E FF 21 FC E9 77 2B 3E :880
D340 30 77 21 13 0E 22 DB E9 3E 04 21 DD E9 77 AF 21 :63F
D350 CF E9 77 C9 CD BC D2 3E EC 21 CE E9 77 21 D6 E9 :AAC
D360 7E DD 21 DF E9 16 01 06 04 0E 00 DD 71 00 FE 00 :5BF
D370 28 08 0E 01 DD 71 01 3D 18 0A 5F 7A 0E 10 81 DD :442
D380 77 01 57 7B 5F 3E 11 90 DD 77 04 7B 0E 0C C3 85 :5BD
D390 DB 0E 03 DD 71 06 C5 01 07 00 DD 09 C1 05 C2 69 :5E4
D3A0 D3 00 00 00 C3 2D D3 3E 01 C3 61 D3 21 D5 E9 34 :6DF
D3B0 00 00 00 00 2A DB E9 00 DB 01 FE FE CA D3 D3 DB :811
D3C0 00 FE BF CA D9 D3 FE FB CA DF D3 FE EF CA E5 D3 :D17
D3D0 C9 00 00 2D 06 01 C3 E8 D3 24 06 02 C3 E8 D3 2C :651
D3E0 06 03 C3 E8 D3 25 06 04 7D FE 00 C8 FE 1A C8 CD :7A6
D3F0 F3 03 7E FE A5 C3 90 DB FE 20 C0 21 DD E9 70 C9 :A43

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D400 05 CA 14 D4 05 CA 16 D4 05 CA 1E D4 25 7C FE 00 :6D0
D410 C0 26 1C C9 2D C9 24 7C FE 1D C0 26 01 C9 2C C9 :721
D420 21 DD E9 46 2A DB E9 22 D1 E9 CD 00 D4 7D FE 00 :913
D430 CA 55 D4 FE 1A CA 55 D4 E5 CD F3 03 7E E1 FE A5 :AA8
D440 CA D0 D4 FE 20 CA 33 D5 DA C0 D9 FE EC CA 27 D8 :B84
D450 FE E9 CA A8 D8 21 00 1C 2B 7C B5 20 FB C9 00 00 :7AE
D460 22 DB E9 16 C8 1E 87 CD D9 D2 2A D1 E9 16 28 1E :821
D470 20 CD D9 D2 C9 00 00 00 21 CF E9 7E 2F 77 21 01 :680
D480 00 FE FF CA 92 D4 1E 00 CD E6 D2 1C 1C 1C 1C 20 :760
D490 F7 C9 1E 00 CD E6 D2 1D 1D 1D 1D 20 F7 C9 00 00 :6B7
D4A0 01 10 27 CD BC D4 01 E8 03 CD BC D4 01 64 00 CD :710
D4B0 BC D4 0E 0A CD BC D4 7D C6 30 12 C9 C5 37 3F 3E :7CC
D4C0 30 ED 42 38 03 3C 18 F9 C1 09 12 13 13 C9 00 00 :4B2
```

パックマン マシン語プログラム

```
D4D0 E5 2A FD E9 23 22 FD E9 7C B5 FE 00 20 08 21 FF :897
D4E0 E9 3E 03 77 E1 C9 11 98 F5 E5 CD A0 D4 E1 11 E8 :9E9
D4F0 03 AF ED 52 D2 01 D5 E1 CD 60 D4 CD 78 D4 C3 2E :985

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D500 D5 21 D3 E9 7E FE 01 CA F7 D4 3E 01 77 3E 20 D3 :8AB
D510 40 E1 CD 60 D4 21 D4 E9 4E 0C 71 21 52 F9 CB 21 :823
D520 CB 21 06 00 09 3E 87 77 CD 55 D4 AF D3 40 21 CE :6DE
D530 E9 35 C9 CD 60 D4 C3 55 D4 E5 D5 2A CC E9 54 5D :A1E
D540 19 19 7D 84 67 11 81 00 19 22 CC E9 7C D1 E1 C9 :713
D550 19 22 28 19 22 28 19 19 18 20 26 18 20 26 18 18 :1E4
D560 19 22 28 19 22 28 19 19 22 22 20 20 1E 1E 19 19 :1EA
D570 E5 AF 5E 0E 40 06 20 57 21 00 10 ED 41 CD 8C D5 :64A
D580 ED 79 CD 8C D5 B7 ED 52 30 F1 E1 C9 D5 16 03 15 :958
D590 20 FD 1D 20 F8 D1 C9 21 50 D5 CD 70 D5 11 6F D5 :899
D5A0 AF E5 ED 52 E1 CA 54 D3 23 C3 9A D5 7A 06 02 FE :97A
D5B0 03 38 02 90 C9 80 C9 DD 66 04 DD 6E 03 DD 7E 06 :6D5
D5C0 FE 01 CA 43 D6 FE 03 CA D6 D5 E5 21 FC E9 7E E1 :AA2
D5D0 FE FF CA 43 D6 C9 C1 DD 7E 01 FE 01 CA E6 D5 3D :A87
D5E0 DD 77 01 00 00 C9 3E 04 DD 77 05 16 28 1E E0 CD :5C2
D5F0 D9 D2 3E 0B DD 77 03 3E 0F DD 77 04 3E 20 DD 77 :6A2

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D600 02 AF DD 77 06 21 0B 0F CD 0B DC C9 DD 7E 00 FE :71C
D610 00 2F DD 77 00 28 04 1E E8 18 02 1E EB CD D9 D2 :650
D620 C9 FD 21 DF E9 FD 56 04 FD 5E 03 E5 AF ED 52 E1 :A18
D630 C8 00 00 11 07 00 FD 19 C3 25 D6 FF 21 CB E9 72 :6FA
D640 C3 B7 D5 C1 DD 56 05 CD AC D5 5F CD 39 D5 E6 03 :9B9
D650 3C BB CA 4B D6 47 5F 00 CD 60 D6 C3 44 D6 00 00 :768
D660 22 D1 E9 CD 00 D4 7D FE 00 CA 9B D6 FE 1A CA 9B :9B0
D670 D6 43 E5 CD F3 03 7E E1 FE E8 CA 9F D6 FE EB CA :BF8
D680 9F D6 FE E9 CA 9F D6 FE 87 CA E4 D8 FE 21 DA BC :C5B
D690 D7 FE A5 CA BC D7 FE EC CA BC D7 2A D1 E9 C9 CD :C98
D6A0 21 D6 CD BD D7 DD 7E 06 FE 01 C3 FB DB 00 7D FE :9CC
D6B0 0C C2 C8 D6 7C FE 06 DA BF D6 FE 18 DA C8 D6 E5 :ACE
D6C0 21 D5 E9 7E E1 E6 03 C8 DD 70 05 E5 CD F3 03 7E :967
D6D0 21 CB E9 56 E1 DD 74 04 DD 75 03 F5 CD 0C D6 F1 :94B
D6E0 DD 5E 02 DD 77 02 2A D1 E9 16 28 CD D9 D2 C9 00 :7F6
D6F0 DD 21 DF E9 16 48 CD 3C D6 E5 21 DB E9 7E E1 DD :A09

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D700 46 03 B8 CA 0C D7 D2 2F D7 C3 1F D7 E5 21 DC E9 :90A
D710 7E E1 DD 46 04 B8 00 00 00 D2 27 D7 C3 37 D7 06 :6E5
D720 01 CD 3F D7 C3 0C D7 06 02 CD 3F D7 C3 44 D6 06 :758
D730 03 CD 3F D7 C3 0C D7 06 04 CD 3F D7 C3 44 D6 58 :7AE
D740 DD 56 05 CD AC D5 BB C8 43 C3 60 D6 DD 21 E6 E9 :A12
D750 16 68 CD 3C D6 E5 21 DC E9 7E E1 DD 46 04 B8 CA :930
D760 68 D7 D2 80 D7 C3 90 D7 E5 21 DB E9 7E E1 DD 46 :ADE
D770 03 B8 DA 78 D7 C3 88 D7 06 01 CD 3F D7 C3 44 D6 :8CD
D780 06 02 CD 3F D7 C3 68 D7 06 03 CD 3F D7 C3 44 D6 :7B6
D790 06 04 CD 3F D7 C3 68 D7 DD 21 F4 E9 16 C8 CD 3C :8B1
D7A0 D6 C3 44 D6 DD 21 ED E9 16 A8 CD 3C D6 CD 39 D5 :9FF
D7B0 1F DA 44 D6 1F DA F9 D6 C3 55 D7 FF D1 DD 7E 06 :9FB
D7C0 FE 01 C2 AD D6 DD 7E 01 3D DD 77 01 FE 10 DA 07 :821
D7D0 D8 16 28 D5 E5 21 D5 E9 7E E1 D1 E6 01 CA 1C D8 :984
D7E0 D5 E5 DD 70 05 CD F3 03 7E E1 DD 74 04 DD 75 03 :8D8
D7F0 D1 F5 1E E9 CD D9 D2 F1 DD 5E 02 DD 77 02 2A D1 :9C4

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D800 E9 16 28 CD D9 D2 C9 FE 00 CA 13 D8 1F DA D1 D7 :9BC
D810 16 E8 C3 D3 D7 AF DD 77 06 C3 C8 D6 2A D1 E9 1E :9D7
D820 E9 CD D9 D2 C9 00 00 E5 DD 21 DF E9 0E 04 C5 DD :989
D830 56 05 CD AC D5 DD 77 05 DD 7E 06 FE 03 CA 59 D8 :85F
D840 3E 01 DD 77 06 DD 66 04 DD 6E 03 16 28 1E E9 CD :640
D850 D9 D2 21 D8 E9 7E DD 77 01 11 07 00 DD 19 C1 0D :73C
D860 C2 2E D8 00 2A FD E9 11 05 00 19 22 FD E9 30 08 :647
D870 21 FF E9 3E 03 77 E1 C9 E5 3E 02 21 CA E9 77 E1 :8BC
D880 C3 E6 D4 FF 21 00 00 11 0A 00 06 08 1F 30 01 19 :42F
D890 EB 29 EB 10 F7 EB 2A FD E9 19 22 FD E9 DA DE D4 :AAE
D8A0 EB 11 B2 FB C3 18 D9 FF CD 60 D4 2A DB E9 CD 21 :A39
D8B0 D6 3E 0C FD 77 03 3E 0C FD 77 04 3E 03 FD 77 06 :614
D8C0 C3 6E DB 00 3E 10 C3 D8 DB 21 CA E9 7E CD 84 D8 :94B
D8D0 21 01 00 1E D0 CD E6 D2 1D 20 FA 21 CA E9 7E 87 :7A5
D8E0 77 C9 00 00 C1 DD 7E 06 FE 01 C2 0E D9 DD 5E 02 :747
D8F0 CD E6 D6 3E 0C DD 77 03 3E 0C DD 77 04 3E 03 DD :6EA

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D900 77 06 3E 10 C3 E3 DB 00 00 00 00 C3 C9 D8 CD AD :72A
D910 D6 21 FF E9 3E 02 77 C9 CD AC D4 2A FD E9 E5 11 :9B2
D920 98 F5 CD A0 D4 E1 11 E8 03 AF ED 52 D8 21 D3 E9 :A4E
D930 7E FE 01 C8 3E 01 77 3E 20 D3 40 21 D4 E9 4E 0C :6A4
D940 71 21 52 F9 CB 21 CB 21 06 00 09 3E 87 77 CD 55 :622
D950 D4 AF D3 40 00 00 00 00 C9 21 CE E9 7E FE 46 CA :7C3
D960 6D D9 FE 8C CA 6D D9 21 C9 E9 AF 77 C9 CD F6 DB :B40
D970 FE FF C8 3E FF 77 21 DA E9 77 21 D9 E9 3E 40 77 :9AC
D980 C9 21 DA E9 7E FE FF C2 B4 D9 CD A8 D9 D0 E5 21 :B9B
D990 D6 E9 5E E1 16 88 CD D9 D2 21 D9 E9 7E 3D 77 FE :A27
D9A0 00 C0 21 DA E9 AF 77 C9 21 0D 0F E5 CD F3 03 7E :7F6
D9B0 E1 FE 21 C9 CD A8 D9 D0 1E 20 16 28 CD D9 D2 C9 :9A4
D9C0 F5 CD 60 D4 F1 CD 84 D8 21 40 00 1E A0 CD E6 D2 :9B4
D9D0 21 DA E9 AF 77 C9 00 00 21 FB E9 7E 3D 77 FE 00 :808
D9E0 C0 23 7E FE FF 2F 77 CA F7 D9 C3 EE DB E6 1F 21 :A50
D9F0 FB E9 77 CD 01 DA C9 3E D0 21 FB E9 77 CD 01 DA :9FE

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
DA00 C9 1E 04 DD 21 DF E9 DD 7E 06 FE 01 28 09 DD 56 :775
DA10 05 CD AC D5 DD 77 05 01 07 00 DD 09 1D 20 E8 C9 :688
DA20 CD AC D3 CD 20 D4 CD F0 D6 CD 4C D7 CD 98 D7 CD :B99
DA30 A4 D7 CD 59 D9 CD 81 D9 CD D8 D9 CD 58 DA 21 CE :B0D
DA40 E9 7E FE 00 CA E0 DA 21 FF E9 7E FE 02 CA 88 DA :A9C
DA50 FE 03 C8 C3 20 DA 00 00 21 D6 E9 3E 20 46 90 67 :701
DA60 00 2E 00 C3 58 D4 00 00 DD 21 DF E9 0E 04 DD 66 :638
DA70 04 DD 6E 03 DD 5E 02 16 28 C5 CD D9 D2 C1 11 07 :6E3
DA80 00 DD 19 0D C2 6E DA C9 21 00 80 CD 58 D4 CD 68 :7A5
DA90 DA 2A DB E9 16 C8 1E 87 E5 CD D9 D2 E1 21 00 40 :8EA
DAA0 CD 58 D4 2A DB E9 3E 20 D3 40 1D E5 CD D9 D2 21 :8F3
DAB0 00 30 CD 58 D4 E1 7B FE 80 C2 AA DA 1E 2A E5 CD :943
DAC0 D9 D2 21 01 00 1E 00 CD E6 D2 1D 1D 1D 1D 1D 20 :521
DAD0 F5 E1 16 28 1E 20 CD D9 D2 21 D4 E9 35 C9 00 00 :7A6
DAE0 21 00 80 CD 58 D4 CD 68 DA 2A DB E9 16 28 1E 20 :713
DAF0 CD D9 D2 06 08 C5 21 53 F3 1E 19 16 08 7E FE 28 :6AB

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
DB00 CA 62 DB FE E8 CA 68 DB 23 23 15 C2 FD DA 01 68 :957
DB10 00 09 1D C2 FB DA 21 00 50 CD 58 D4 C1 05 C2 F5 :7A4
DB20 DA 21 D6 E9 7E FE 1F CA 2C DB 3C 77 21 D7 E9 7E :938
DB30 21 D0 E9 46 B8 CA C0 DB 21 D8 E9 7E 06 0F 90 DA :91C
DB40 56 DB C3 9B DB AF 21 D7 E9 77 21 D0 E9 34 34 21 :8D4
DB50 D8 E9 3E 50 77 C9 21 D8 E9 3E 01 77 21 D7 E9 34 :83C
DB60 C9 00 3E E8 77 C3 08 DB 3E 28 77 C3 08 DB FD 7E :80A
DB70 02 FE A5 CA 7E DB FE EC CA 7E DB C3 C4 D8 21 CE :B23
DB80 E9 35 C3 C4 D8 DD 71 03 0E E0 DD 71 02 C3 91 D3 :933
DB90 CA FB D3 FE E9 CA FB D3 C3 F8 D3 77 C3 5C DB 26 :C3C
DBA0 26 26 33 26 26 1E 1E 26 26 26 33 26 26 33 33 26 :284
DBB0 26 26 33 26 26 20 1D 1B 1D 20 26 20 26 26 26 00 :218
DBC0 AF D3 51 21 9F DB CD 70 D5 11 BE DB AF E5 ED 52 :9FD
DBD0 E1 CA 45 DB 23 C3 C6 DB FD 77 01 3E E0 FD 77 02 :95B
DBE0 C3 C9 D8 DD 77 01 3E E0 DD 77 02 C3 07 D9 E6 0F :8C5
DBF0 06 10 80 C3 EF D9 21 C9 E9 7E C9 C1 C2 04 DC 3A :8D8

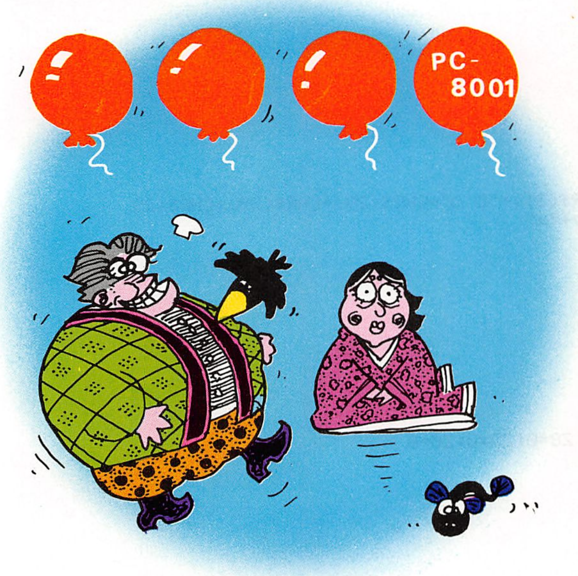
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
DC00 D5 E9 1F D0 FD 7E 02 DD 77 02 C9 E5 21 CB E9 56 :959
DC10 E1 C3 0C D6 FF FF FF FF FF FF FF FF FF FF FF FF :E7A
```

PC-8001 BASIC

クレイジーバルーン

CRAZY BALLON

渡辺　卓也



TVゲーム『CRAZY BALLOON』のPC版です.

ルール

左右に揺れる風船をスタートからゴールまで，割らないように持っていきます．失敗は3度しか許されません．

遊び方

RUNさせると，キーの説明が画面に表示されます．使用するキーはテンキーの[2]（下），[4]（左），[6]（右），[8]（上）だけです．次に[スペース]キーを押してください．コースが現われ，左上にtimeが表示されます．

ゲームはスタート位置にある風船をゴールまで持っていくのだが…



ここまでくるのが初めのうちは大変！



ここまでくると後は簡単そうにみえるが，難関はまだ続く



3つ目の風船が割れてゲーム・オーバー



右下の風船をうまく操って，できるだけ速くゴールまで持っていくのですが，あせるとグラフィック・キャラクタにぶつかって，割れてしまいます．

風船は4個あります．ゴールインするか，風船が4個とも割れるとゲーム・オーバーです．時間の制限はありません．

もう一度やりたいときは［スペース］キーを押してください（**注**：スタートに近いところで風船が割れると次の風船も現われて，すぐ割れてしまうことがあります）．

プログラムについて

プログラムはＢＡＳＩＣで書いてあります．コメントが入れてあるので，全体の流れはリストを見ればわかると思います．

230行で色の指定をＲＮＤ(1)＊6＋2で行なっているのは，青色が見えにくいためです．カラーモニタを使用している人はＲＮＤ(1)＊7＋1にして良いと思います．

730行で風船がグラフィック・キャラクタにぶつかったかどうかをＶ-ＲＡＭの値を読んで判定しています．770行も同じです．220行のＢＡが風船の数です．

最後に

ＢＡＳＩＣでもスピードはそれほど遅いとは思いませんが，ＰＣでは音楽が出せないのが残念です．

このゲームはゲーム・コーナーでタイトーの『ＣＲＡＺＹ ＢＡＬＬＯＯＮ』を見て，その夜作ったものですが，楽しんでもらえたでしょうか．実物では風が吹いたり，コースが変わったり，100円玉が必要だったりします．ＤＡＴＡ文やＬＩＮＥ文の数値を変えると，色々なコースができるので試してください．

我がＰＣのソフトはＭＺのソフトに数の点で負けているので，ＰＣのユーザーのみなさん，I/OにもっとＰＣのソフトを発表しましょう．

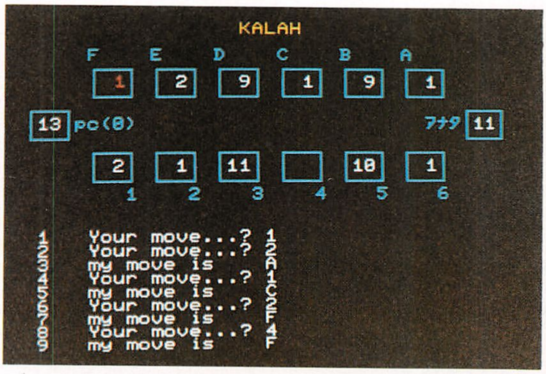
CRAZY BALLOON プログラム・リスト

```
10 '***********************
20 '***                 ***
30 '*** CRAZY BALLOON   ***
40 '***                 ***
50 '*** Copyright 1980 ***
60 '***       by        ***
70 '***  T.WATANABE     ***
80 '***                 ***
90 '***********************
100 A$="X^X"+CHR$(29)+CHR$(29)+CHR$(29)+CHR$(31)+"O O"+CHR$(29)+CHR$(29)+CHR$(29
)+CHR$(31)+"!-!"+CHR$(29)+CHR$(29)+CHR$(31)+"X"
110 '
120 '    ｹﾞｰﾑ ﾉ ｾﾂﾒｲ
130 '
140 CONSOLE0,25,0,1:WIDTH40,25
150 LINE(0,0)-(39,24)," ",BF
160 GOSUB 1020
170 '
180 ' ｶﾞﾒﾝ ﾂｸﾘ
190 '
200 LINE(0,0)-(39,24)," ",BF
210 Z=0:Z1=0:Z2=0:Z3=0:Z4=0:Z5=0:Z6=0:Z7=0:Z8=0:Z9=0:Z0=0
220 D=1 :M=1:BA=4
230 COLOR RND(1)*6+2
240 LINE(23,14)-(38,16),"●"
250 COLOR RND(1)*6+2
260 LINE(10,15)-(20,12),"♥"
270 COLOR RND(1)*6+2
280 LINE(8,  8)-(12,12),"♣"
290 COLOR RND(1)*6+2
300 LINE(4 ,23)-( 9,23),"◢"
310 LINE(19,23)-(28,23),"◢"
320 COLOR RND(1)*6+2
330 LINE(14,2 )-(17,6 ),"■"
340 COLOR1
350 LINE(0,0)-(39,24),"■",B
360 COLOR RND(1)*6+2
370 LINE(35,8 )-(39,8 ),"◢",B
380 COLOR RND(1)*6+2
390 LINE(25,2 )-(29,5 ),"◤",B
400 RESTORE
410 READ X,Y:IF X<>0 THEN LOCATE X,Y:COLOR RND(1)*6+2:PRINT "◆";:GOTO 410
420 DATA  14,17,20,22,25,16,1,17,2,1, 8,16,26,7,25,6,2,10,5,2,23,13,0,0
430 COLOR7
440 LOCATE 33,24:PRINT "START";:LOCATE34,0:PRINT "GOAL";
450 COLOR7
460 '
470 ' ｹﾞｰﾑ ｽﾀｰﾄ
480 '
490 TIME$="00:00:00"
500  XX=34:Y=18
510 '
520 ' ﾒｲﾝ
530 '
```

```
540  GOSUB 670
550 M=M+D :IF M=5  OR M=1  THEN D=-D
560 I=INP(0):J=INP(1)
570 IF I=251 THEN Y=Y+1:GOTO 610
580 IF I=239 THEN XX=XX-1:GOTO 610
590 IF I=191 THEN XX=XX+1:GOTO 610
600 IF J=254 THEN Y=Y-1
610 LOCATE 5,0:PRINT "time";TIME$;
620 IF XX>33 AND Y<3 THEN 960
630 GOTO 540
640 '
650 ' フウセン ヲ カク
660 '
670 GOSUB 910
680 X=XX-(3-M)
690 Z=X*2+Y*120+&HF300
700 Z1=Z+2:Z2=Z+4:Z3=Z+120
710 Z4=Z+124:Z5=Z+240:Z6=Z+242:Z7=Z+244
720 Z8=Z+362
730 IF PEEK(Z)+PEEK(Z1)+PEEK(Z2)+PEEK(Z3)+PEEK(Z4)+PEEK(Z5)+PEEK(Z7)<>224  THEN
830
740 POKEZ,238:POKEZ1,148:POKEZ2,239:POKEZ3,136:POKEZ4,151:POKEZ5,239:POKEZ6,128:
POKEZ7,238:POKEZ8,240
750 POKEZ9,32 :POKEZ0,32
760 Z9=Z-SGN(M-3)*2+482:Z0=Z+608-M*2
770 IF PEEK(Z9)+PEEK(Z0)<>64  THEN 830
780 POKEZ9,240:POKEZ0,135
790 RETURN
800 '
810 ' ハレツ
820 '
830  LOCATEX-1,Y:PRINT A$;:BEEP
840 BA=BA-1:IF BA<1 THEN 860
850 GOTO 500
860 LOCATE 15,22:PRINT "GAME OVER";
870 IF INKEY$=" " THEN 200   ELSE870
880 '
890 ' フウセン ヲ ケス
900 '
910 POKEZ,32 :POKEZ1,32 :POKEZ2,32 :POKEZ3,32 :POKEZ4,32 :POKEZ5,32 :POKEZ6,32 :
POKEZ7,32 :POKEZ8,32
920 RETURN
930 '
940 ' ゴール イン
950 '
960 LOCATE 10,24:PRINT "GOAL IN !!";
970 BEEP
980 GOTO 870
990 '
1000 ' セツメイ
1010 '
1020 H=6:S=2:Y=5
1030 FOR H=H TO 26 STEP S
1040 XX=H:FOR M=1 TO 5:GOSUB 670:NEXT M:FOR M=5 TO 1STEP-1:GOSUB670:NEXT M
1050 IF H=10 THEN LOCATE 9,10:PRINT "B";:LOCATE8,6:PRINT "C";
1060 IF H=12 THEN LOCATE11,10:PRINT "A";:LOCATE10,6:PRINT "R";
1070 IF H=14 THEN LOCATE13,10:PRINT "L";:LOCATE12,6:PRINT "A";
1080 IF H=16 THEN LOCATE15,10:PRINT "L";:LOCATE12,8:PRINT "●";
1090 IF H=18 THEN LOCATE17,10:PRINT "O";:LOCATE14,6:PRINT "Z";
1100 IF H=20 THEN LOCATE19,10:PRINT "O";:LOCATE16,6:PRINT "Y";
1110 IF H=22 THEN LOCATE21,10:PRINT "N";
1120 NEXT
1130 '
1140 ' セツメイ ブン
1150 '
1160 LOCATE 8,12:PRINT " フウセン ヲ ウマク ゴール ニ イレテ クダサイ。          フウセン ハ 4コ アリマス。"
;
1170 LINE(7,5)-(22,11),"♥",B
1180 LOCATE 15,15:PRINT "8";:LOCATE15,16:PRINT "|";:LOCATE13,17:PRINT "4─┼─6";:L
OCATE15,18:PRINT "|";:LOCATE15,19:PRINT "2";
1190 '
1200 ' バー ヲ オス
1210 '
1220 LOCATE 10,22:PRINT "PUSH BAR";
1230 FOR I=0 TO 30:IF INKEY$=" " THEN200 ELSE NEXT:S=0:H=26:GOTO 1030
1240 '
1250 ' プログラム カンリョウ。 ゴクロウサン!
1260 '
1270 '  1980年 6月 24日
1280 '
1290 '
1300 '
```

COLOR PHOTO

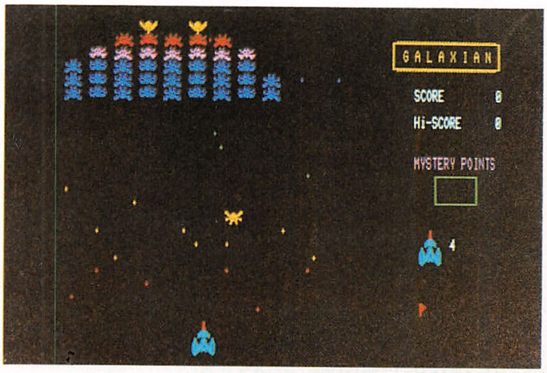
KALAH GAME



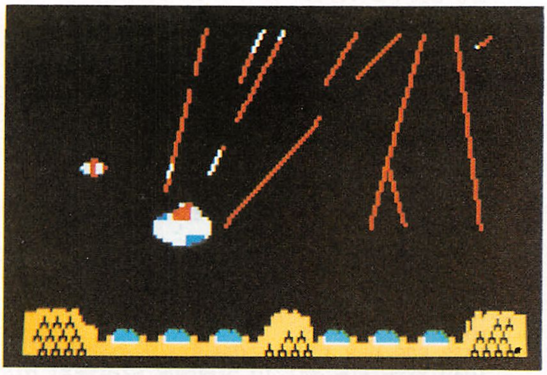
グラフィック麻雀



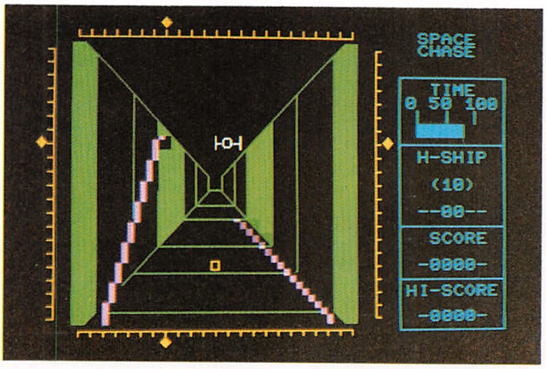
PCGギャラクシアン



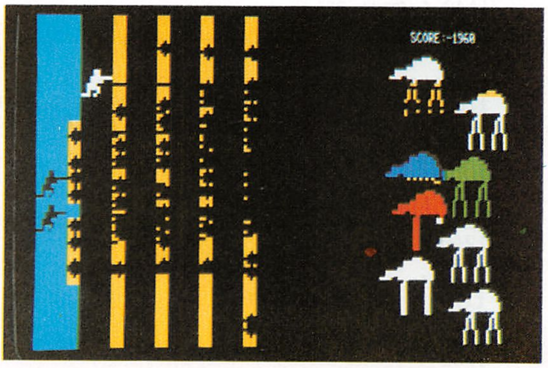
ミサイル・コマンダー



スペース・チェイス



スターウォーズ





タイトル文字・イラスト：©手塚プロ・火の鳥プロ・東宝(東宝提供)映画紹介p.121

このゲームは映画『火の鳥』からヒントを得て作りました．あくまでもヒントですから，映画の筋とは多少違います．

ゲーム内容は大宇宙を飛び回る火の鳥を宇宙船スペース・シャークに乗って生け捕るというゲームです．すなわち，火の鳥を捕らえる量を競うゲームです．

遊び方

1 キーの説明

ビーム砲は前と後の2方向で，フルキーボードの1が後，2が前です(図1 a)．

シャークは上下左右に移動できます．キーはテンキーボードの5の上下左右のキーです(図1 b)．このとき，スペースキーを押しながら移動キーを使うと，動くスピードが上ります．

2 火の鳥の捕え方

まず，ビーム砲で火の鳥を気絶させます（火の鳥は，不死鳥ですから死にません）．すると，%が@に変わります．

図1　操作キー

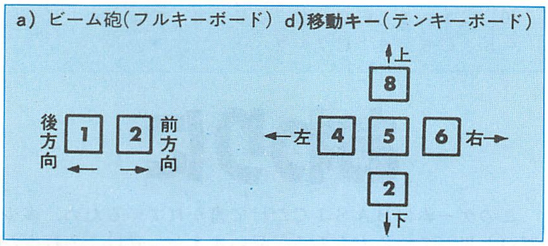
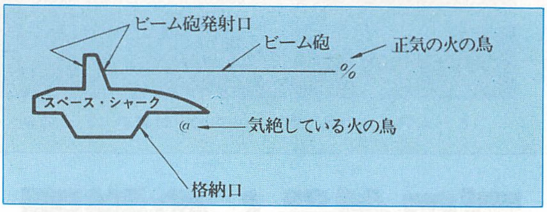


図2　スペース・シャークの機能



そこで，シャークを移動させ火の鳥を格納入口から格納します(図2)．このとき，格納はそっと行なってください（すなわちスペースキーを押さないで)．そうしないとうまく入らないことがあります．

うまく格納できると 100点～ 500点の間の得点が得られます．

以上が捕え方です．

3 その他

火の鳥は始め2羽ですが，1面終わらすごとに2羽ずつ増えます．4面目以後は8羽のままです．

シャークが正気の（気絶していない）火の鳥にある程度以上近寄ると，火の鳥はシャークに炎を放ち，シャークは爆発します．すると，シャークは1台減ることになります．注意してください．

気絶した火の鳥はある程度時間が経つと正気に戻ります．また，気絶している火の鳥にビーム砲を当てても正気に戻ります．後者の場合，次に気絶したときは気絶している時間が短かくなるので注意してください．

画面の最上部の表示は左から，トップ・スコア，今のスコア，シャークの残り数（今使っているシャークを除く）です．

なお，4面目を終わらせたときのみシャークが1台増えます．

映画火の鳥登場人物
©手塚プロ・火の鳥プロ・東宝

ゴドー



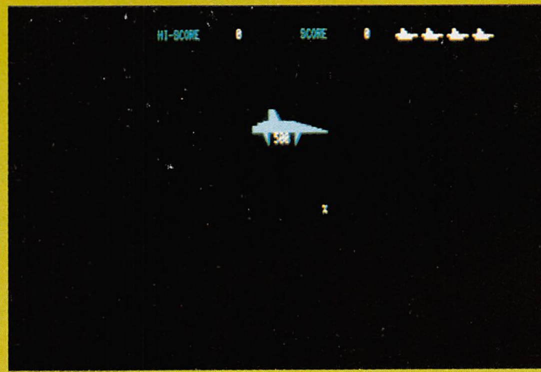
試験管ベビーとして生まれ，幼児期はもっぱら1人でオートマチック育児システムを持つドームハウスで将来の宇宙ハンター要員として育てられた．5才ぐらいの頃，宇宙科学センターから育児ロボット，オルガが送られてきて，のちオルガに一切の世話を受け成人する．



●ガンキャリア



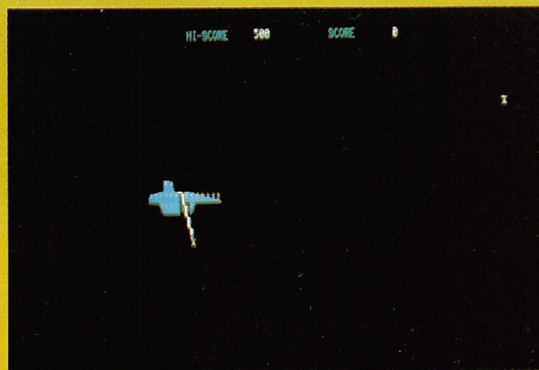
RUN直後の画面，**f·1** キーを押すとゲーム・スタート



スペース・シャックへ火の鳥をうまく格納して500点加算．



ビーム砲が火の鳥に命中．中央の@は気絶した火の鳥



火の鳥に近づきすぎると攻撃される．

プログラムの説明

プログラムの説明といっても，そうむずかしいプログラムでないのでここでは簡単に説明します(図3，表1)．

まず，火の鳥ゲームのデモですが，これは必ず入れましょう（筆者からのお願い！）．

行番号３２０で行番号５６０にジャンプしていますが，これはスペース・シャークを表示するためだけのものです．キーを調べるのには，ＩＮＰを使っています．

Ｕ％（Ｉ，Ｊ）は画面上のＩ，Ｊ座標の火の鳥の状態を表わすデータが入っています．たとえば，火の鳥が正気なときは－1，気絶したときは－2，火の鳥がそこにいないときは０が入っています．行番号６００では，うまく格納したかどうかは，この座標を調べています．

レーザー砲は，まずレーザーが発射されたとき火の鳥に当たるかどうかを先に調べてから，レーザーの表示をしています．この方法が一番早いようです．

火の鳥の移動は一度に全部ではなく，ここ（行番号８４０～９９０）を通るごとに１羽ずつ動かします．このため，何かの原因(スペース・シャークがその上を移動したなど）で火の鳥が消えると次にその火の鳥の番がくるまで消えたままになってしまいます．それでは困るので，１１１０行～１１５０行で改めて火の鳥を表示します．

火の鳥の射程内にスペース・シャークがいるかどうかは，その火の鳥が移動したときに調べます．１０１０行，１０２０行の±10や±５を変えれば火の鳥の射程が変わります．

表１　プログラムで使った主な変数

U%(I，J)	I，J位置（画面上）の状態
FX(I)	火の鳥 I のX座標
FY(I)	火の鳥 I のY座標
F(I)	火の鳥 I の気絶してからのカウント
FS	火の鳥の始めの数
X	スペース・シャークのX座標
Y	スペース・シャークのY座標
D	スペース・シャークの残り台数
FI	火の鳥をつかまえた数
SC	スコア
HC	ハイ・スコア

10行のＣ１～Ｃ７の変数は色の指定用なのでカラーモニタを使っている方は，次のように変更してください．

```
10　C1＝1：C2＝2：C3＝3：C4＝4：C5
＝5：C6＝6：C7＝7
```

おわりに

このゲームはＢＡＳＩＣだけで書かれているため，多少遅いと思っている方もあると思いますが，私はこの程度が

オルガ



宇宙科学センターから送られてきた宇宙ハンター育成用育児ロボット．人間的な反応を見せはするが，ことばはややカタコトで感情にとぼしく，全く人間と同様というわけではない．

ロック



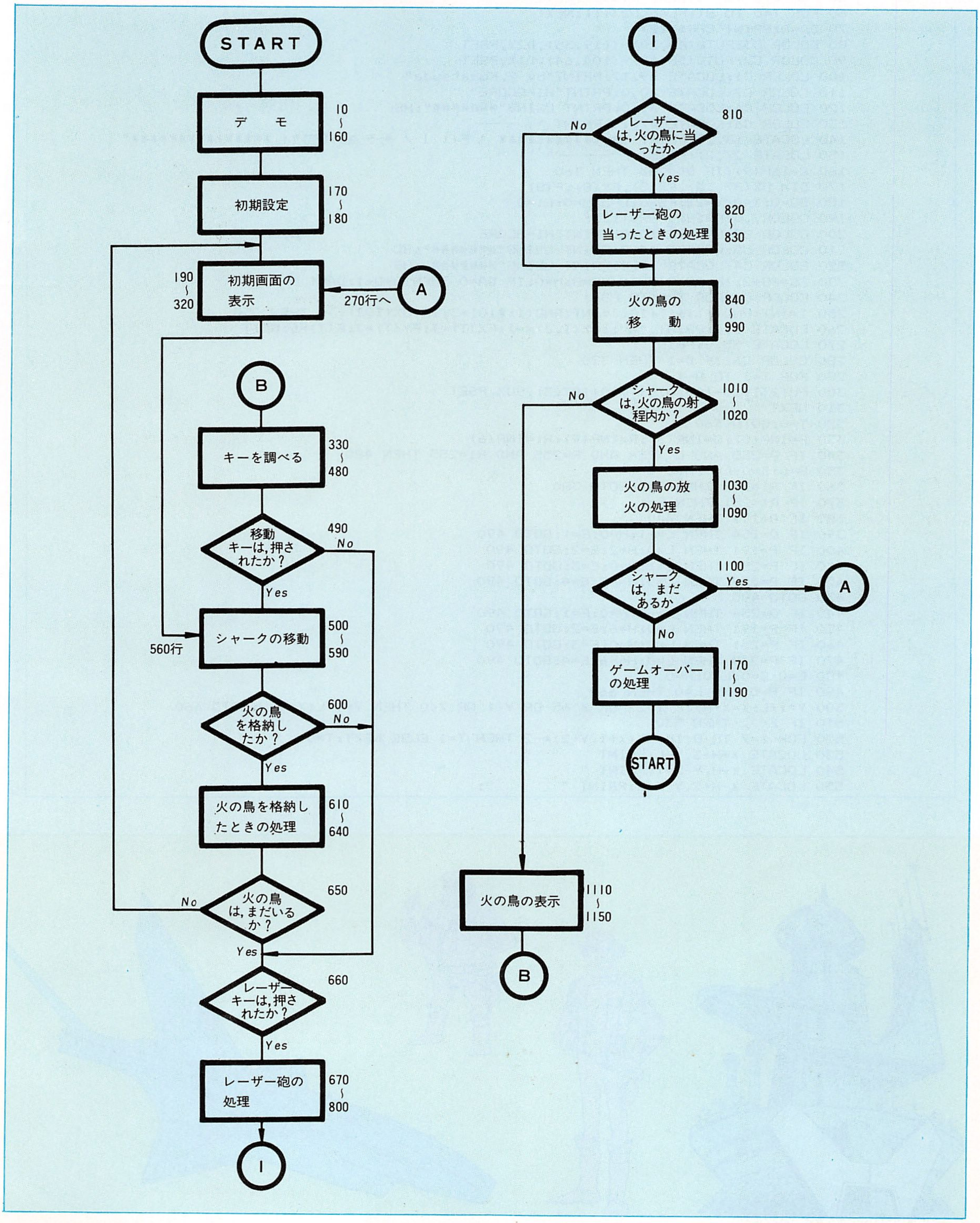
知能指数300という超英才少年．性格は冷徹功利的．徹底した合理主義の精神のもとにあくなき権力への野心を秘め，人を人とも思わない．エリート指向，エリート意識を強くもっている．しかし，彼は彼なりにある種の使命感を持って人類の将来について考えるところはある．ただし，それは常に自分の権力欲を満足させることとウラハラになっているのだ．

ちょうどいいと思っています.(負け惜しみではない).

最近，ゲームのネタが出つくした傾向がありますが，考えればまだまだあるはずです．私はヒマがあればなにかネタがないか考えています．特に学校の行き帰りの電車の中，つまらない授業のとき（何の授業かは言えませんが察しはつくでしょう）などです．

みなさんもがんばりましょう．

図3　火の鳥ゲームのフローチャート



レナ



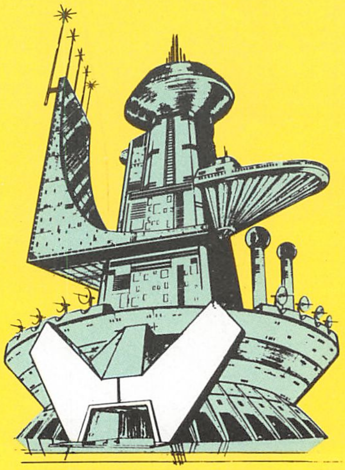
ハイソサエティの権力者の娘として生まれ，生まれながらのスーパーエリートとして何不足のない育ちである．しかし，この頃のエリート階級の合理主義的，非人間的な生活意識になじめず，純朴で人間味たっぷりのゴドーに出会い心ひかれ恋仲になる．が，それを引きさかれ，父親のイート卿の歓心をひくため，レナに結婚を申し入れ許婚になっていたロックに強引に入籍させられると，ロックに忠実な妻となってしまう．あまり主体性のない育ちのいい娘といったところだ．

サルタ

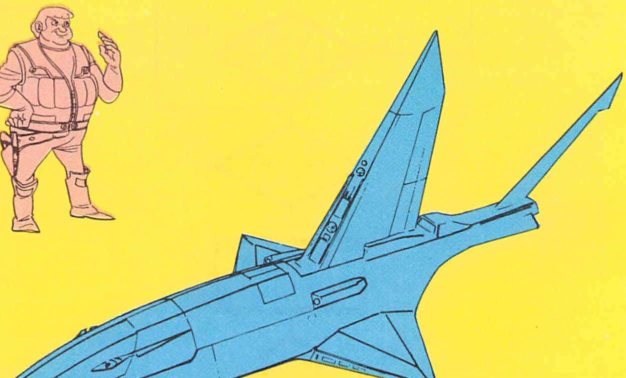
地球物理をはじめ，かなり広い範囲の自然科学を専門とする科学者．もともとが専門職としてスーパーエリートではなくそれにつぐ階級に属し，マントル対流エネルギー開発委員長という要職にあった．が，その計画の危険性に気づき，上司のロックに計画の中止を上申したため，委員長の地位を追われ，マントル対流の建設現場の監督機関であるマントル対流研究所の所長として労働キャンプで働かされている．

火の鳥ゲーム プログラム・リスト

```
10 C1=7:C2=7:C3=4:C4=7:C5=7:C6=7:C7=5
20 CONSOLE ,,0,1:COLOR7,,1:WIDTH80,25
30 DIM SU5(3),D1%(79),D2%(81)
40 FOR T=0 TO 3:READ SU%(T):NEXT
50 FOR T=0 TO 79:READ D1%(T):NEXT
60 FOR T=0 TO 81:READ D2%(T):NEXT
70 DA=0:PRINT CHR$(12)
80 COLOR C3:PUT@(40,20)-(119,35),D2%,PSET
90 COLOR C2:PUT@(55,40)-(104,64),D1%,PSET
100 COLOR C1:LOCATE 39,17:PRINT"by Y.Komatsuda"
110 COLOR C7:LOCATE 30,0:PRINT"HI-SCORE"
120 COLOR C4:LOCATE 39,0:PRINT USING"#######";HS
130 COLOR C6:LOCATE 27,19:PRINT " ╭──────╮"
140 LOCATE 13,20:PRINT "************** | F･1 | ノ キーヲ オシテクダ゛サイ ****************"
150 LOCATE 27,21:PRINT " ╰──────╯"
160 Q=INP(9):IF Q<>253 THEN 160
170 DIM U%(77,25),FX(8),FY(8),F(8)
180 SC=0:X=0:Y=22:FS=0:D=5:M=0:F1=0
190 COLOR,,0:PRINT CHR$(12);
200 COLOR C7:LOCATE 10,0:PRINT"HI-SCORE                          SCORE"
210 COLOR C5:LOCATE 43,0:PRINT USING"#######";SC
220 COLOR C4:LOCATE 19,0:PRINT USING"#######";HS
230 FS=FS+2:IF FS>8 THEN FS=8:M=0:IF DA=0 THEN D=D+1:DA=1
240 COLOR C2:FOR T=0 TO FS-1
250 I=INT(RND(1)*61)+10:J=INT(RND(1)*10)+3:IF U%(I,J)<>0 THEN 250
260 LOCATE I,J:PRINT "%";:U%(I,J)=-1:FX(T)=I:FY(T)=J:F(T)=0:NEXT
270 LOCATE 55,0:PRINT "                         ";
280 COLOR C6:IF D=1 THEN 320
290 FOR T=1 TO D-1
300 PUT@(T*10+100,0)-(T*10+107,3),SU%,PSET
310 NEXT
320 T=0:GOTO 560
330 P=INP(0):Q=INP(1):R=INP(9):R1=INP(6)
340 IF P=255 AND Q<>254 AND R=255 AND R1=255 THEN 480
350 E=0:S=0:L=0:H=0
360 IF R1=253 THEN S=2:GOTO 380
370 IF R1=251 THEN S=1
380 IF R=191 THEN 440
390 IF Q=254 THEN L=-1:H=0:E=1:GOTO 490
400 IF P=191 THEN L=0:H=2:E=2:GOTO 490
410 IF P=251 THEN L=1:H=0:E=3:GOTO 490
420 IF P=239 THEN L=0:H=-2:E=4:GOTO 490
430 GOTO 490
440 IF Q=254 THEN L=-2:H=0:E=1:GOTO 490
450 IF P=191 THEN L=0:H=6:E=2:GOTO 490
460 IF P=251 THEN L=2:H=0:E=3:GOTO 490
470 IF P=239 THEN L=0:H=-6:E=4:GOTO 490
480 E=0:S=0:L=0:H=0
490 IF H=0 AND L=0 THEN 660
500 Y=Y+L:X=X+H:IF Y>22 OR X>65 OR Y<1 OR X<0 THEN Y=Y-L:X=X-H:GOTO 660
510 IF E<>2 THEN 530
520 FOR I=7 TO 8:IF U%(X+I,Y+2)=-2 THEN T=1 ELSE NEXT:T=0
530 LOCATE X-H+3,Y-L:PRINT "  "
540 LOCATE X-H,Y-L+1:PRINT "         "
550 LOCATE X-H+2.Y-L+2:PRINT "     ";
```







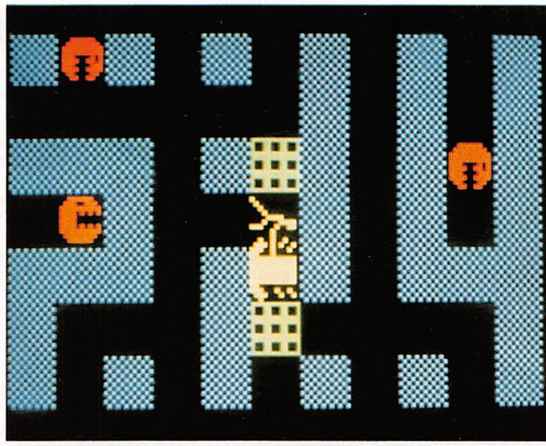
ブラックジャック　マントル対流エネルギー計画の建設労働キャンプの所長．もともとは医者であったがエリート中心の社会のありかたに疑問を持ったため反体制派と見られ，労働キャンプに所長として送りこまれている．一方，キャンプ内に診療所を開き，囚人以外の連中の診療に当っている．知的で豪胆な男．

ヒゲオヤジ　涙星と呼ばれるレグルス23番惑星にただ一人世捨て人のように住んで宇宙生物たちの世話や研究をしているへんくつな老人．サルタの古い友人でもある．人嫌いであるが宇宙生物たちにはたよりになるおやじである．動物園のベテラン飼育係の退職後の姿といったところ．

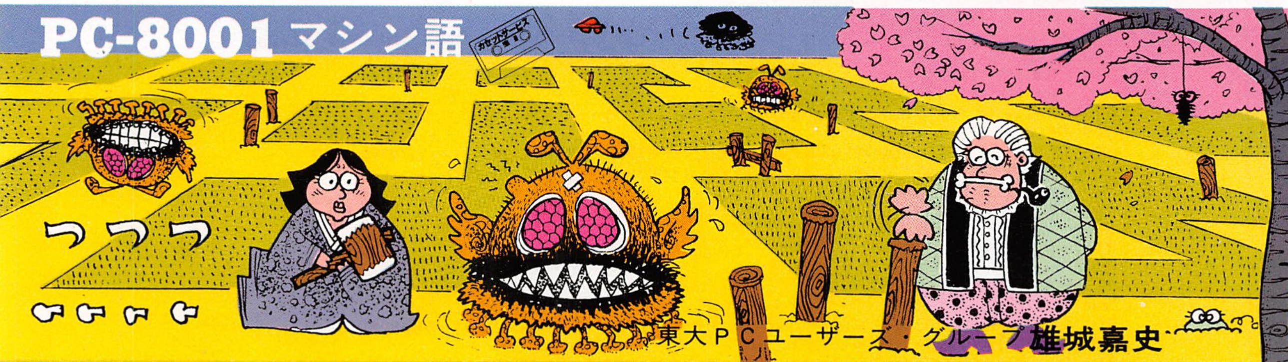
ピンチョ(宇宙人)　ピンチョ星の宇宙人．地球に連れてこられてペットとして飼われている．一応人間なみの知能をもち，人間の行動についての批判力もある．どういうわけか地球人が好きで飼い主のほんのちょっとの愛情みたいなものにも，ものすごく反応して喜ぶ．気のいい犬ころみたいな宇宙人で，やたらにそうじするくせがある．

```
560 COLOR C7
570 LOCATE X+3,Y:PRINT    "█◣"
580 LOCATE X,Y+1:PRINT "██████████▄▄▁▁"
590 LOCATE X+2,Y+2:PRINT "◥████◤";
600 IF T=0 THEN 660
610 V=INT(RND(1)*5)*100+100
620 COLOR C2:LOCATE X+3,Y+2:PRINT V;:U%(X+I,Y+2)=0:T=0:SC=SC+V:F1=F1+1
630 FOR I=0 TO 100:BEEP1:FOR I1=0 TO 5:NEXT:BEEP0:NEXTI
640 COLOR C5:LOCATE 43,0:PRINT USING"#######";SC:S=0
650 IF F1=FS THEN F1=0:X=0:Y=22:GOTO 190 ELSE 560
660 IF S=0 THEN 840
670 COLOR C6
680 IF S=2 THEN O=2:GOTO 740 ELSE O=5
690 BEEP1:EL=78
700 FOR I=0 TO FS-1:IF FY(I)<>Y THEN 720
710 IF FX(I)<X+5 OR EL<FX(I) OR U%(FX(I),FY(I))=0 THEN 720 ELSE EL=FX(I):J=I
720 NEXT:GOTO 780
730 IF EL=78 THEN CO=0 ELSE CO=U%(EL,Y)
740 BEEP1:EL=0
750 FOR I=0 TO FS-1:IF FY(I)<>Y THEN 770
760 IF FX(I)>X+2 OR EL>FX(I) OR U%(FX(I),FY(I))=0 THEN 770 ELSE EL=FX(I):J=I
770 NEXT
780 LINE(EL,Y)-(X+O,Y),"-"
790 LINE(EL,Y)-(X+O,Y)," ":BEEP0
800 IF EL=78 THEN CO=0 ELSE CO=U%(EL,Y)
810 IF CO=0 THEN 840
820 IF CO=-1 THEN COLOR C3:LOCATE EL,Y:PRINT "@";:U%(EL,Y)=-2:GOTO 840
830 IF CO=-2 THEN COLOR C2:LOCATE EL,Y:PRINT "%";:U%(EL,Y)=-1
840 M=M+1:IF M=FS THEN M=0
850 ON INT(RND(1)*4+1) GOTO 860,870,880,890
860 H=0:L=-3:GOTO 900
870 H=6:L=0:GOTO 900
880 H=0:L=3:GOTO 900
890 H=-6:L=0
900 COLOR C2
910 IF FX(M)+H<10 OR FX(M)+H>70 OR FY(M)+L<3 OR FY(M)+L>21 THEN 1110
920 IF U%(FX(M),FY(M))=-1THEN 940
930 IF U%(FX(M),FY(M))=-2THEN F(M)=F(M)+1:GOTO 980 ELSE 1110
940 IF U%(FX(M)+H,FY(M)+L)<>0 THEN 1110
950 LOCATE FX(M),FY(M):PRINT " ":U%(FX(M),FY(M))=0
960 FX(M)=FX(M)+H:FY(M)=FY(M)+L
970 LOCATE FX(M),FY(M):PRINT "%":U%(FX(M),FY(M))=-1:GOTO 1000
980 IF F(M)=10-FS THEN F(M)=0:U%(FX(M),FY(M))=-1 ELSE 1110
990 LOCATE FX(M),FY(M):PRINT "%"
1000 A=X+6:B=Y+1
1010 IF A>FX(M)+10 OR A<FX(M)-10 THEN 1110
1020 IF B>FY(M)+5 OR B<FY(M)-5 THEN 1110
1030 LINE(FX(M)*2,FY(M)*4)-(A*2,B*4),PSET:LOCATE FX(M),FY(M):PRINT "%";
1040 BEEP:BEEP
1050 LINE(FX(M)*2,FY(M)*4)-(A*2,B*4),PRESET:LOCATE FX(M),FY(M):PRINT "%";
1060 LOCATE X+3,Y:PRINT    "**"
1070 LOCATE X,Y+1:PRINT "**************"
1080 LOCATE X+2,Y+2:PRINT "*******";
1090 FOR I=0 TO 200:BEEP1:BEEP0:NEXT
1100 D=D-1:IF D=0 THEN 1160 ELSE S=0:GOTO 270
1110 FOR M1=0 TO FS-1:MT=U%(FX(M1),FY(M1))+3
1120 ON MT GOTO 1130,1140,1150
1130 COLOR C3:LOCATE FX(M1),FY(M1):PRINT "@";:GOTO 1150
1140 COLOR C2:LOCATE FX(M1),FY(M1):PRINT "%";
1150 NEXT:GOTO 330
1160 COLOR C6:LOCATE 30,12:PRINT "***** GAME OVER *****"
1170 IF SC>HS THEN HS=SC
1180 FOR I=0 TO 5000:NEXT
1190 ERASE U%,FX,FY,F:GOTO 70
1200 '***** ﾃﾞｰﾀ ********
1210 DATA 32 , 3076 , 16127 , 0
1220 DATA 1250 , 0 , 0 , 3552 , 0 , 0 , 992 , 0
1230 DATA 0 ,-3104 , 0 , 0 , 1984 , 3 , 0 , 32640
1240 DATA 0 , 0 ,-256 , 28 , 0 ,-512 , 1 , 0
1250 DATA-2048 , 479 , 0 ,-4096 , 15 , 0 ,-8192 , 6015
1260 DATA 0 ,-16384 , 127 , 0 ,-16384 , 1023 , 0 , 0
1270 DATA-17409 , 0 , 0 , 4094 , 0 , 0 ,-28680 , 11
1280 DATA 0 ,-16 , 6 , 14320 ,-32 , 0 , 992 ,-106
1290 DATA 119 ,-17216 ,-253 , 31 , 32704 ,-512 , 255 ,-786
1300 DATA-1 , 4095 ,-8177 ,-1 ,-1 , 1919 ,-2 ,-1
1310 DATA 13311 ,-64 ,-1 , 1023 ,-512 ,-1 ,-8193 , 2
1320 DATA 1280 , 0 , 0 , 96 ,-2116 , 30 , 768 , 0
1330 DATA 96 ,-27484 , 18 , 25368 , 0 , 4095 ,-31728 , 8
1340 DATA 25368 , 0 , 3075 , 16904 , 4 , 8976 , 0 , 4095
1350 DATA 16900 , 2 , 13104 , 0 , 3075 , 16956 , 30 , 4896
1360 DATA 0 , 4095 , 0 , 0 , 768 , 0 , 3 , 1280
1370 DATA 0 , 768 , 0 , 16383 , 1288 , 256 , 1920 , 0
1380 DATA 3 , 252 , 384 , 3264 , 1920 , 16383 , 66 , 192
1390 DATA 6240 , 2112 , 12288 , 64 , 96 , 12336 , 4384 , 14043
1400 DATA-928 , 48 , 24600 , 4384 , 14043 , 48 , 536 ,-16372
1410 DATA-30400 , 13165 , 24 , 2040 ,-32761 ,-31357 , 6436 , 8
1420 DATA 2048 , 0
```

『ＦＩＲＥ』シリーズ第３弾をお届けします．今回の目標は，迷路作制をマシン語で行なうこと，マイコン・ゲームでは余り見られなかった上下左右スクロールの実現および利用，また，ＰＣに適した色制御手法の考察などでした．
また，今回はプログラムの細分構造化を試み，アセンブラをまったく使っていません．
ゲームは『平安京エイリアン』のスーパーバージョンを目指し，かつ，迷路脱出ゲームの面白さも取り入れて作制したオリジナル・ゲームです．
『全プログラマーの敵，バグをハンマーで叩きつぶし，デバッグの迷路から脱出しよう！！』



PC-8001マシン語



東大ＰＣユーザーズ・グループ　雄城嘉史

バグ・ファイア BUG FIRE!

ゲームの説明

プログラムは＆ＨＣ100から＆ＨＥ53Ｂの9Ｋバイト強の領域を使用します．もちろん，16Ｋシステムでも動きます．
『全機械語のゲームはＲＵＮさせるのが面倒で困る』という方のために，このゲームはＢＡＳＩＣとリンクすることもできます．その場合は**リスト１**の１行のみをキーインしてからロードしてください．モニタからＲＵＮさせるためには**GE3CC** RETURN で走ります．
ＲＵＮすると**写真１**のような画像が出現します．キーボードの説明の他，音を出すか出さないかを尋ねてくるので，Y キーまたは N キーで答えてください．次にゲームの説明が必要か否かを尋ねてくるので同様に答えてください．
ここで，画面は**写真２**の状態に転じます．これは中間報告で，各面の終了ごと，またはバグ（エイリアン）にプレイヤーが喰い殺されるたびにこの状態となります．
さて，ゲームの目的は迷路内を走りまわるバグをたたきつぶし，迷路を脱出することにあります．バグはプレイヤーを見つけると容赦なく喰い殺します．これに対し，プレイヤーはストッパーとハンマーで対抗します．
ストッパーとはプレイヤーが自分の**後側**に作る人工の壁で，プレイヤーの走ろうとしている方向の**逆側**に出現しま

図１　キーボードの説明

❶テンキーによる走行と向きの制御

8（↑）
4（←）　6（→）
2（↓）

❷テンキーと共に押すことにより有効となるキー

SHIFT　ハンマー（テンキーにより示す方向に振り降ろす）
X　ストッパー（テンキーとは逆の方向に作る）

❸ゲーム中止，もしくはモニタまたはBASICに戻る．

STOP

リスト１

```
10  CLEAR0, &HC0FF:
    DEFUSR0=&HE390:A
    =USR0 (A)
```

写真１　キーの説明と音の指定

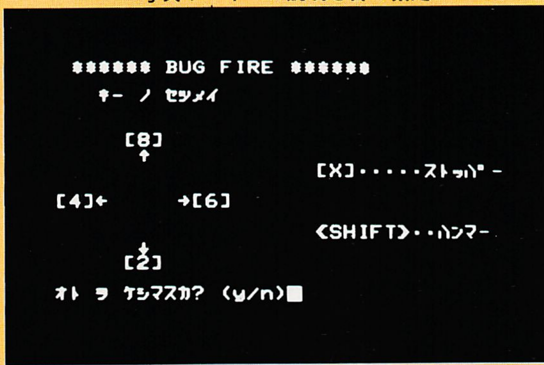


写真２　ゲームの中間報告

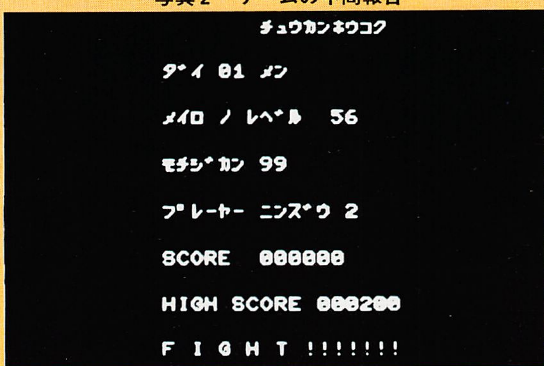


写真３　ゲーム開始直後の画面



写真４　左方向へ走っているところ（ただし，人間はいつも中央にいる）



写真５　バグに喰われてしまった

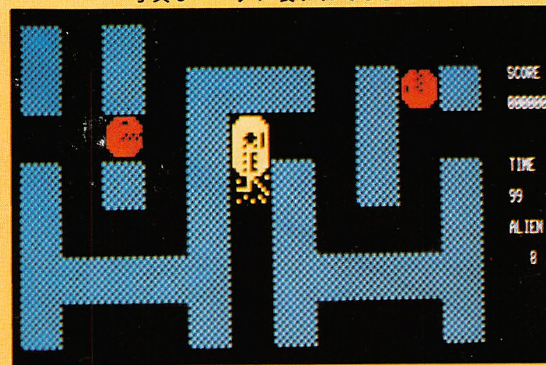


写真６　ハンマーでバグを叩きつぶす．しかし，…

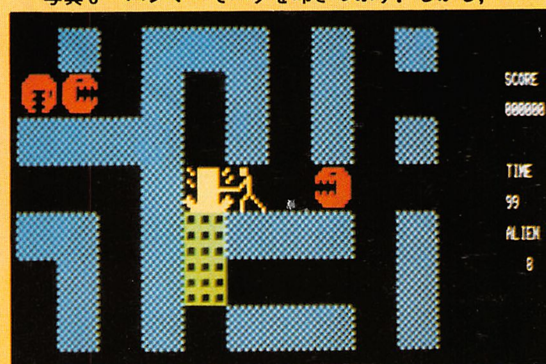
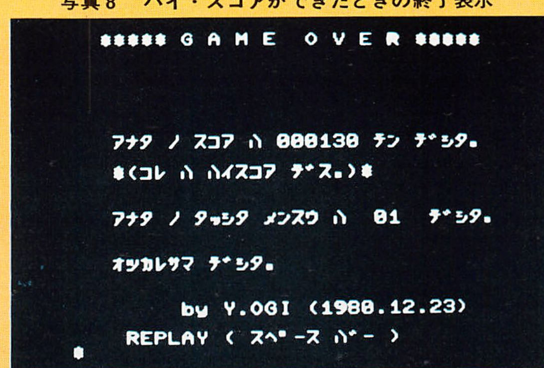


写真７　バグと格子状のストッパー



写真８　ハイ・スコアができたときの終了表示



す．バグはストッパーにぶつかるとしばらくの間動かなくなるので，このときに襲いかかってハンマーが叩きつぶしてください．

ストッパーは一定時間たつと消えてしまいます．バグの動かない時間はそのほぼ倍です．**図1**にキーボードの説明を示します．なお，ストッパーは迷路中に同時に6個までしか存在できません．

さて，**写真2**の状態が数秒続くと画像は**写真3**の状態に転じます．画面中央の人間がプレイヤーで，このゲームではプレイヤーは常に画面中央に位置し，周囲の画像が動くことにより移動を表現します．

迷路の構成および難易度は変化し，特に構成は各面スタート時におよそ3～6秒かけて作制するもので，毎回異なります．

図2に迷路の説明を示します．迷路の大きさは30×30ですが，画面にはこのうち横11，縦7の範囲しか見えません．

写真4に左向きにプレイヤーが走っている状態を，**写真5**にバグに喰われた瞬間を，**写真6**にハンマーでバグを叩きつぶした瞬間を，そして**写真7**に走り回るバグとストッパーを示します．

写真7の状態で，ストッパーは緑の格子状のものであり，バグは目と口のついた赤い球体として示されているはずです．

バグは各面スタート時に8匹おり，この8匹すべてをやっつけることにより出口が開くとともに出口のすぐ前のところから再度8匹のバグが出現します．

1度開いた出口はプレイヤーが殺され新しいプレイヤーに代わっても開いていますが，バグはプレイヤーが交代すると8匹すべて生き返りますし，8匹すべてを殺すたびにバグは再び出口前からウジャウジャとわいて出ます．

ですからプレイヤーは出口が開いたら適当な時期にバグを残したまま脱出してください．もし意地になってバグつぶしを続けていると，やがて時間がきてプレイヤーは死にます．また，ハンマーの空振りは10点減点です．このとき，得点が負になってもプレイヤーは死にます．バグを残したまま迷路を脱出するというのはどうも気分が良くありませんが，ま，お許しを．

さて注意すべき点としては，『後に何かあってはストッパーはおけない』のです．したがって，バグとの追いかけっこの最中，ストッパーで逃げようとしてもバグが近すぎるとストッパーは出ません．また気絶していないバグを叩くためにはバグの隣を叩いてください．

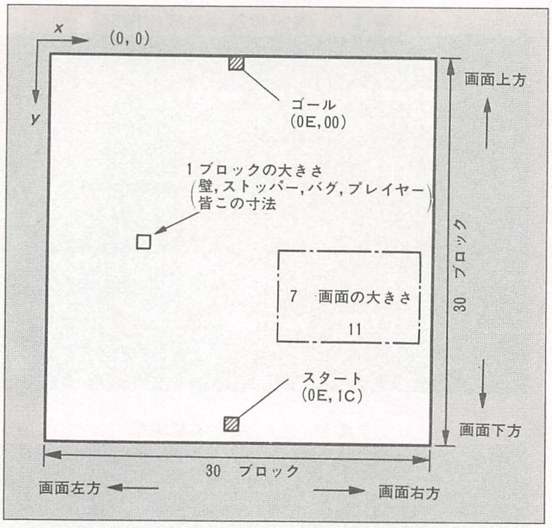
他にくどいですがストッパーは迷路内に6個までです．これを忘れると，あわてふためくことになります．また，ストッパーもハンマーも方向キーを押していないと使えません．

画面について

画面はプレイヤーの移動とともに上下左右にスクロールし，迷路内のプレイヤーを中心とした11×7の領域を表示しています．

画像転送とアトリビュート・エリア制御を別々に行なうため，移動時に一瞬色ずれがほんのわずか見えますが，この方法を取った理由は後述します．また，移動の感じが出てかえって良いという人もいるので気にしないでください．画像の移動幅は各方向変わらず4ドットで，まぁアナログ的に見えます．

図2　迷路構成と画面



プレイヤーもバグも各方向2通りの画像を使用し，プレイヤーは両手両足を振って走り，バグは口を開閉しながら移動します．プレイヤーは普通白色をしていますが，ハンマーを使ったり喰われたりすると黄色になります．

ハイ・スコアについて

写真8にハイ・スコアを出した場合のゲーム・オーバーの様子を示します．ハイ・スコアは＆ＨＣ546～＆ＨＣ548の3バイトに10進数で登録されています．ゲーム開始時に得点は0点に更新しますが，ハイ・スコアはむろん更新しません．

テープをロードする際，できるだけロード時間を短縮するためワーク・エリアはセーブ範囲に入れない場合があり，このときハイ・スコア領域に奇妙な定数が残り得ることが考えられます．

この範囲は起動時には＆Ｈ00が書き込まれている領域なのでスイッチ・オンしてからすぐロード，ＲＵＮする場合は良いのですが，他のプログラムを扱った後でこのゲームを行なう場合にはハイ・スコアの管理はユーザーの責任で実行してください．

プログラムの説明

プログラムおよびデータ・エリアは＆ＨＣ５Ｃ０から＆ＨＥ53Ｂの８Ｋ強の領域を占めます．前述したとおり，このゲーム作制にあたりアセンブラの類はいっさい使用していません．

一度Ｚ80の命令を覚えてしまえば，アセンブラなど使うのはかえって面倒なためもありましたが，一応プログラミング手法の実験のつもりでした．徹底的な細分構造化を自己に強制するためだったのです．

マシン語プログラムのバグ捜しが大変なことは皆さんご存知だと思いますが，すべてハンド・アセンブルで行なう

表1　主要サブルーチン

アドレス	ルーチン名	説　明
&HCC00	SECCON	迷路内座標により指定された点のデータを出力
&HCC60	VECTOR	移動による新座標を出力
&HCCF4	MAZSEC	迷路作制時に進行可能か否かを判断
&HCD1C	GOVCHK	戦闘時の進行許可
&HCD38	SRND	乱数発生サブルーチン
&HCD5C	MZVSCN	迷路作制時進行方向決定
&HCD88	ALSCAN	バグ進行方向決定
&HCE10	MAKMAZ	迷路作制（完全迷路）
&HCE94	MAZCTL	ループ付加も含めた迷路作制最上位サブルーチン
&HCEAC	LEFLOW	画面移動のうち左方走行に対応する画像シフト
&HCEC8	RIGROW	同右
&HCEE0	UPLOW	同上
&HCEF8	DOWNLW	同下
&HD00C	LEFT	画面端処理も含めた左方走行
&HD013	RIGHT	同右
&HD01A	UP	同上
&HD021	DOWN	同下
&HD15C	PUTWRO	画像位置・幅調整機能付き画像処理サブルーチン
&HD1E0	OGHMDA	画面中央制御
&HD21C	ALMOVE	バグ移動制御（画像移動）
&HD254	STALCN	気絶したバグの制御
&HD278	DEADCN	プレイヤー死亡制御
&HD2A4	ALSECN	バグ移動制御
&HD338	ALWRLT	バグ消去・表示
&HD38C	VPSLV	迷路座標を画面内座標に変換
&HD3D0	STWLCN	ストッパー制御
&HD450	AMWRIT	プレイヤー消去・表示
&HD480	PHAPIC	画像移動上位サブルーチン
&HD490	PHASE	画像制御上位サブルーチン
&HD4DC	STWLPT	ストッパー作制
&HD528	GETHM	バグ殺害制御
&HD580	OUTHM	ハンマー空振り制御
&HD5BC	HMCONT	ハンマー関係制御
&HD5F0	INPUT	キー入力サブルーチン
&HD64C	AMMOVE	プレイヤー移動
&HD6A0	MAINLW	下位主制御部
&HD82C	PREPO	初期化サブルーチン(その1)
&HDADC	CLCNTL	色の移動制御
&HDBEC	LINYMK	アトリビュート完成サブルーチン
&HDCA0	COLMOV	色制御最上位ルーチン
&HDD1C	PRVRAM	ゲーム開始時画面作制サブルーチン
&HD920	GMVPR0	ゲーム初期化サブルーチン(その2)
&HDE58	STRING	文字表示サブルーチン
&HDEC8	REPSUB	中間報告サブルーチン
&HE052	RUNLOW	最上位サブルーチン
&HE390	MAIN	メイン・ルーチン
&HE3CC	MACHP0	Gコマンド用ポインタ

表2　主要ワーク・エリアの説明

アドレス	説　明
&HC100 〜 &HC48F	迷路領域．30×30の寸法がとれる．各バイトが1ブロックに対応し，第7ビットが通常壁，第6ビットがストッパー，第5ビットがバグに対応する．
&HC490 &HC491	SECCON入力迷路座標．SECCON出力は&HC492〜&HC497に置かれる．
&HC498 〜 &HC4A8	迷路作制時穴を掘るための座標，アドレス，迷路情報エリア．
&HC4A9 〜 &HC4AC	SRND出力およびポインタ
&HC4B0 〜 &HC4B4	画像位置，消すか書くか，データ・ポインタ入力．画像処理に使用するバッファは&HC4B5〜&HC4C8まで．
&HC4C9 〜 &HC4CC	プレイヤーデータ．
&HC4CD	ストッパー設置スイッチ．
&HC4CE	ハンマー・スイッチ．
&HC4CF	入力禁止フラグ（キー入力）
&HC4D0 〜 &HC4D5	ストッパーの状態表．
&HC4D6 〜 &HC4DD	バグの状態表．
&HC4DE 〜 &HC501	ストッパーデータ・エリア（各6バイト×6）
&HC502 〜 &HC541	バグデータ・エリア(各8バイト×8)
&HC542 〜 &HC544	得点（HIGH　SCOREと共に10進数6桁）
&HC546 〜 &HC548	HIGH　SCORE．

となると徹底した準備が必要となり，かつ個々のサブルーチンはジャンプなどの関係から短くなり，解析が楽になってバグが減り，しかもデバッグも楽なのです．

表1で主要サブルーチンを，**表2**で主要ワーク・エリアを説明します．フローチャートは少々複雑になりすぎるので省略することをお許しください．

PCは画面制御にアトリビュート・エリアという特殊な手法を使っています．しかもその制御はなかなか難しく，処理時間もかかります．ではアトリビュート・エリアという画面制御法は好ましくないものなのでしょうか？　そんなことはありません．

たとえば，このゲームをRGB3枚のV-RAMを持つシステムで実行すると相当遅くなるはずです．扱う画像の位置や幅がある程度規則正しいものならばアトリビュート・エリアの制御は簡単となり，画像によっては3組のV-RAMを使う色制御手法などより処理速度は大となるでしょう．

このゲームでは扱うグラフィック画像はすべて1辺が12の倍数ドットからなる四辺形になっており，色制御を必要とするものに関してはその位置は完全に管理されています．

色制御は6×3＝18キャラクタからなる正方形を単位として実行されます．走り回るバグは赤色をしていますが，実はバグが赤いのではなく道が赤いのです．画面上で空虚な黒に見える道は実はアトリビュート・エリアでは赤色，(&H58)が宣言されており，そこに書かれるものはそのままではすべて赤くなってしまうのです．

プレイヤーやストッパーはその色を再宣言しているのです．このゲームでは道もヌル・キャラクタ18個の集合として扱っており，青い壁と何ら差はありません．このように，迷路側に色をつけることにより色制御は迷路構造と対応して実行可能となり，動き回るバグの色を定めるのにいちいちアトリビュート・エリア内のスキャン（ROM内ルーチンでは&H0622）などを行なう必要はなくなります．

迷路の構成単位でもある18キャラクタの集合が画面内のどこにあるかは迷路構造とプレイヤーの存在位置および後

述するＰＨＡＳＥＸ，ＰＨＡＳＥＹにより簡単に求められます．

x幅は６キャラクタと一定で，重なることはなく隣接しているこれらの色を制御することはアトリビュート・エリアで簡単に，しかも早く実行できるのです．ＰＨＡＳＥＸ，ＰＨＡＳＥＹは各々左端および上端のブロックが３分のいくつ見えているかを示す変数と考えてください．

アナログ的に見せるため，移動は迷路構成単位（ブロック）の縦横幅の1/3を単位として行なわれ，色制御もこれに対応するためにこれらの変数が必要となります．色制御はどうしても各ブロックの画面内座標を管理することが必要で速いといっても，バイト数ではむしろ多いV-RAM内データのブロック転送と比較しうる時間を要します．

画面内データのシフトと対応するアトリビュート・エリア制御を同時に実行することにより上下左右スクロールを行なうと，特に左右スクロールで画面のちらつき，画像の歪みが生じ見づらくなってしまいます．

このため，両者を分離し画像データを転送してから色制御に入るため，前述の色ズレが生じます．しかし，これは解決可能です．

また，各ブロックの座標管理といっても必要なデータは前述のＰＨＡＳＥＸ，ＰＨＡＳＥＹの２変数値のみで，処理の速度は早く，前述の画面のちらつきもＴＶの走査線との同期が運の悪い形になったとき，ときどき見えるだけなので余り気にならず，むしろその方法が良かったかも知れません．

なお，上下左右スクロールはブロック転送ＬＤＩＲ，ＬＤＤＲにより画面内データをシフトし，画面端に迷路構造より判断した値を書き込んで実現しています．

迷路作製アルゴリズムは単純で，壁の中に穴を掘っていく方法をとっています．入口から出口まで正しい行き方が１通りしかない迷路はおよそ３秒で完成しますが，これに各面ごとに定まった数のループを与えるために最大３秒ほどよけいに時間を要します．

計６秒というのはマシン語ソフトにとっては大変な時間で，多少の短縮は可能でしょうが，大幅な短縮のためには迷路作制アルゴリズムの変更が必要だと思います．なお，中間報告中『迷路のレベル』とはループ数のことです．





変更点

各面の迷路難易度は＆ＨＤＦ78～＆ＨＤＦ87までに10進数で，＆ＨＤＦ88～＆ＨＤＦ97までに対応する16進数でテーブル化されています．これを変更すると迷路難易度が変化します．

ただし，＄01～＄90の間の数にしてください．また，迷路難易度すなわちゲームの難易度ではないのでご注意を．ループの多い迷路と，ループの少ない迷路では戦法がまるで異なるのです．

最後に

少なくとも，前２作『スター・ファイア』，『スタートレック・ファイア』よりも面白いと友人達から好評のこの『バグ・ファイア』ですが，改良すべき点，改良可能な点は多いと思います．

このゲームはえらく短期間に完成し，しかもいまのところバグは少ないので（わざと残したものもあります．出口近くではご注意！），恐怖のフルハンド・アセンブル作制法もまんざらではないようです．上下左右スクロールおよびその色制御手法を使えば『ラリーX』，『クレージークライマー』などの類似ゲームを作ることは簡単です．

しかし，解像度，バック・カラーの点から上下スクロールだけでクレージークライマーを作るのはなかなかきついお話ですが，どなたか作って見てください．話は変わりますが，I/O’81年２月号に発表されている東大ＴＳＧ会員の方が作った『ミサイル・コマンダー』は見事です．

ちなみに『バグ・ファイア』のハイ・スコアは今のところ筆者の10,257点，『スタートレック・ファイア』のハイ・スコアは88,620点です．これらの記録を破った方はなかなかの根性人間と言えましょう．私の場合，疲れてへたり込みました．

卒論が多忙なため，次作の予告は難しいのですが，フライト・シミュレータ＝ＰＣか，フルマシン語高度戦略ウォー・シミュレーション・ゲームを作ってみたいと考えております（前者はほとんどできているのですが，後者の方が面白そうですし……）．それではこの辺で．

BUG FIRE! プログラム・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
C5C0 C1 C1 C1 C1 C1 C1 C1 C1 C1 C2 C2 C2 C2 C2 C2 C2 :C17
C5D0 C2 C2 C3 C3 C3 C3 C3 C3 C3 C3 C4 C4 C4 C4 C4 C4 :C34
C5E0 00 1E 3C 5A 78 96 B4 D2 F0 0E 2C 4A 68 86 A4 C2 :710
C5F0 E0 FE 1C 3A 58 76 94 B2 D0 EE 0C 2A 48 66 66 66 :7B6

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
C600 00 FF 01 00 00 01 FF 00 9F F9 00 00 A5 A5 06 03 :4EB
C610 80 EC 0A A0 CE 08 F0 F9 8A A8 FF 0F 10 73 77 77 :886
C620 37 01 06 03 80 EC 4E EA CE 08 F0 F9 DF FA FF 0F :88B
C630 10 73 77 77 37 01 06 03 80 EC AE EA CE 08 F0 FF :77B
C640 0F 09 09 09 10 73 77 77 37 01 06 03 80 EC AE EA :4E0
C650 CE 08 F0 FF BF BD BD 0D 10 73 77 77 37 01 06 03 :6BD
C660 80 EC EE EE CE 08 F0 FF 15 51 9F 0F 10 73 05 50 :7F9
C670 37 01 06 03 80 EC EE EE CE 08 F0 FF 5F FB 9F 0F :856
C680 10 73 57 72 37 01 06 03 80 EC AE EA CE 08 90 90 :687
C690 90 F0 FF 0F 10 73 77 77 37 01 06 03 80 EC AE EA :744
C6A0 CE 08 D0 DB DB FB FF 0F 10 73 77 77 37 01 0E 22 :73E
C6B0 36 4A 5E 72 86 9A 02 0A 12 1A 22 2A 32 3A DE E4 :522
C6C0 EA F0 F6 FC 0C 1C 1C 2A 2A 2A 2A 2A 20 47 20 41 :5AA
C6D0 20 4D 20 45 20 20 4F 20 56 20 45 20 52 20 2A 2A :322
C6E0 2A 2A 2A 0A 0A 0A 0A 0A FF FF FF FF FF FF FF FF :8A8
C6F0 F0 88 C8 F0 88 F8 F0 31 84 F0 88 88 02 03 00 01 :85B
```

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
C700 06 03 00 00 E0 06 00 00 00 A4 F1 21 24 00 40 44 :34D
C710 12 11 0F 00 06 03 00 00 E0 06 C0 00 70 11 F1 11 :364
C720 11 00 00 44 74 11 11 01 06 03 00 00 40 08 88 88 :24D
C730 60 46 5E 44 87 00 00 00 10 12 00 07 06 03 00 44 :245
C740 0C 00 0E 00 60 46 4F 44 47 0C 00 00 47 04 00 03 :1F4
C750 06 03 E0 00 48 08 00 00 00 E1 22 7A 62 06 11 11 :340
C760 10 02 00 00 06 03 C0 00 20 E2 00 00 30 E2 22 F2 :403
C770 62 06 00 70 00 30 22 00 06 03 00 00 60 0E 00 00 :1A1
C780 00 42 12 1F 4A 00 00 F0 11 21 44 04 06 03 00 0C :23C
C790 60 0E 00 00 00 11 11 1F 11 07 10 11 11 47 44 00 :184
C7A0 06 03 00 00 E0 06 00 00 80 24 F1 21 84 00 00 00 :329
C7B0 0F 0F 00 00 28 3C 00 14 50 64 78 8C A0 A0 0D 0A :3A5
C7C0 0A 1C 1C 1C B1 C5 C0 20 C9 20 BD BA B1 20 CA 20 :6CF
C7D0 FF FF FF FF 0D 0A 0A 1C 1C 1C 2A 28 BA DA 20 CA :741
C7E0 20 CA B2 BD BA B1 20 C3 DE BD A1 29 2A FF FF FF :A33
C7F0 9F F9 8F F8 02 02 9F F9 88 F8 BE C4 CA D0 D6 DC :B09

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
C800 E2 E8 EE F4 06 06 00 00 00 00 00 00 00 F0 FF FF :6A6
C810 FF FF 00 30 33 F3 33 33 00 00 00 78 00 00 30 43 :4A5
C820 56 44 1A 11 00 00 00 00 10 42 0C 03 00 00 80 66 :20C
C830 00 00 00 88 88 88 00 00 00 00 F0 21 84 88 88 FF :53C
```

BUG FIRE!／プログラム・リスト

```
C840 FF FF 00 00 80 24 01 E1 00 00 00 FF FF FF 00 00 :681
C850 06 06 42 08 00 00 00 00 88 58 22 6A C2 0C 00 00 :290
C860 1E 00 00 00 CC CC CF CC 0C 00 FF FF FF 0F 00 00 :768
C870 00 00 00 00 00 00 00 0C 03 00 00 88 88 88 00 00 00 :1A7
C880 66 08 00 00 00 00 00 FF FF FF 88 88 48 12 0F 00 00 :4E4
C890 00 00 FF FF FF 00 00 00 1E 10 42 08 06 2C 52 78 :471
C8A0 0A 0A 0A 0D 1C 1C 1C B1 C5 C0 20 C9 20 C0 AF BC :5E9
C8B0 C0 20 D2 DD BD B3 20 CA 20 20 FF FF 20 20 C3 DE :908
C8C0 BC C0 A1 0A FF FF FF FF 0D 0A 0A 1C 1C 1C B5 C2 :80F
C8D0 B6 DA BB CF 20 C3 DE BC C0 A1 FF FF 0D 0A 0A 0A :921
C8E0 1C 1C 1C 1C 1C 1C 1C 1C 62 79 20 59 2E 4F 47 49 :341
C8F0 FF FF FF FF 0C 1A 27 36 44 08 0A 0C 00 00 00 00 :4E1
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
C900 06 06 20 04 42 00 72 CE F0 FF FF FF FF FF 30 3B :808
C910 33 3F B3 FF 10 20 01 0F 31 01 30 43 56 44 1A 11 :3CE
C920 00 00 00 00 10 42 0C 03 00 00 80 66 00 28 04 EE :261
C930 EE EE 40 02 00 00 F0 21 44 54 44 FF FF FF 20 01 :729
C940 80 24 01 E1 00 72 CE FF FF FF EC 27 06 06 42 08 :72C
C950 00 00 00 00 88 58 22 6A C4 0C 80 8C F0 80 04 08 :4C4
C960 FF CD FC CC DC 0C FF FF FF FF FF 0F 73 4E 00 42 :A89
C970 20 04 0C 03 20 04 EE EE EE 40 82 00 66 08 00 00 :451
C980 10 02 FF FF FF 44 45 44 12 0F 00 00 72 CE FF FF :73B
C990 FF EC 27 00 1E 10 42 08 B8 58 D8 00 06 06 00 00 :47E
C9A0 00 00 00 00 00 80 EC EF 8C 00 E4 FF FF B7 08 00 :688
C9B0 00 31 E1 06 69 00 00 00 F0 11 00 00 80 24 01 E1 :408
C9C0 00 00 01 02 04 08 CD 40 D9 21 C4 C6 CD 58 DE 21 :5C4
C9D0 BE C7 CD 58 DE 21 44 C5 CD BD 5E 21 43 C5 CD BD :94D
C9E0 5E 21 42 C5 CD BD 5E 21 08 CB CD 58 DE CD B8 E0 :8CA
C9F0 21 A0 C8 CD 58 DE 21 64 C5 CD BD 5E 21 BC C8 CD :930
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
CA00 58 DE 21 C8 C8 CD 58 DE 21 DC C8 CD 58 DE 21 18 :8EB
CA10 CB CD 58 DE C9 FF 73 FF 3B 13 FF 0C 10 F3 CE 00 :932
CA20 06 06 80 42 20 40 20 00 80 21 84 1F 51 00 C0 FE :4A1
CA30 15 5F EF 0C F0 F7 C5 5F FF 0F F0 F8 8D D8 FF 0F :9E3
CA40 10 73 77 77 37 01 0C 03 80 EC AE AA EA EE EE 8C :7CE
CA50 02 48 82 04 F0 FF 9F 90 F6 D4 D4 F4 56 C4 80 02 :91C
CA60 10 73 77 77 77 77 77 13 20 30 00 02 06 06 80 EC :4B3
CA70 EE EE CE 08 F0 FF 1B B1 1F 0F F0 FF FA A3 EF 0F :A25
CA80 30 F7 FA A8 7F 03 00 8A F8 21 84 01 00 04 02 04 :57D
CA90 42 01 0C 03 40 28 84 20 C8 EE EE AE AA EA CE 08 :71A
CAA0 20 08 4C 65 4F 4D 4D 6F 09 F9 FF 0F 20 00 03 02 :466
CAB0 31 77 77 77 77 77 37 01 00 00 00 00 00 00 00 00 :2BC
CAC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
CAD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
CAE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
CAF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
CB00 00 00 00 00 00 00 00 00 00 20 C3 DD 20 C3 DE BC C0 :4FD
CB10 A1 FF FF FF 8F 88 CC 0C 20 28 31 39 38 30 2E 31 :706
CB20 32 2E 32 33 29 FF FF FF 3A CA C4 FD 96 00 4F F2 :887
CB30 34 CB ED 44 FE 07 D0 79 ED 44 C6 06 4F 3A CB C4 :893
CB40 FD 96 01 47 F2 49 CB ED 44 FE 05 D0 78 06 00 C5 :828
CB50 ED 44 C6 04 4F 21 F8 DA 09 6E 26 DB C1 09 3E 58 :715
CB60 77 C9 AF CD C8 E4 21 0E 00 CD 8C D3 22 B0 C4 3E :897
CB70 01 32 B2 C4 21 A8 D3 CD 5C D1 C9 00 00 00 00 00 :608
CB80 00 00 FF 7F 77 F7 FF 00 00 FF 0F 00 00 00 00 00 :4F9
CB90 FF EF EE EE EE FF 0F 05 05 FF 0F 00 F0 FF FF 0F :9DB
CBA0 00 F0 FF FF 0F 00 F0 FF 11 00 00 10 11 EE 0E 00 :61B
CBB0 E0 EE 08 32 48 72 97 0C 1A 27 36 44 06 06 1E 09 :453
CBC0 24 0E 00 0C 03 18 09 24 0B 06 06 1E 06 24 0B 0C 03 :0FF
CBD0 1E 09 2A 0B BC C2 C8 CE 06 2C 52 78 02 28 4E 74 :558
CBE0 22 48 6E 94 D8 DC E0 D8 FF FF 00 FF 00 00 FF FF :9D3
CBF0 FF 00 FF FF 00 FF FF FF FF FF FF 00 00 00 00 00 :9F6
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
CC00 3A 91 C4 FE 1E D2 54 CC 4F 06 00 21 C0 C5 09 7E :71F
CC10 CD D3 D7 6E 67 3A 90 C4 FE 1E D2 54 CC 4F 09 22 :862
CC20 92 C4 7E 32 94 C4 E6 C0 CA 3C CC CB 7F CA 48 CC :9FE
CC30 3E B8 32 95 C4 21 A5 A5 22 96 C4 C9 3E 58 32 95 :78E
CC40 C4 21 00 00 22 96 C4 C9 3E 98 32 95 C4 21 9F F9 :744
CC50 22 96 C4 C9 3E F8 32 95 C4 21 00 00 22 96 C4 C9 :76C
CC60 CD 90 D8 CB 27 4F 06 00 21 00 C6 09 4E 23 46 C9 :5EC
CC70 E5 CD 60 CC E1 7C 80 67 7D 81 6F C9 2A 98 C4 CD :9AB
CC80 70 CC 22 9E C4 CD 70 CC 22 A3 C4 C9 00 00 CD 7C :864
CC90 CC 2A 98 C4 22 90 C4 CD 00 CC 2A 92 C4 22 9B C4 :862
CCA0 3A 94 C4 32 9D C4 2A 9E C4 22 90 C4 CD 00 CC 2A :7EA
CCB0 92 C4 22 A0 C4 3A 94 C4 32 A2 C4 C9 CD 8E CC 2A :920
CCC0 A3 C4 22 90 C4 CD 00 CC 2A 92 C4 22 A5 C4 3A 94 :84F
CCD0 C4 32 A7 C4 C9 00 00 00 FE 01 FA E0 CC FE 1D F8 :8E2
CCE0 3E 01 32 A8 C4 E1 E1 C9 7C CD D8 CC 7D CD D8 CC :A43
CCF0 C9 00 00 00 CD BC CC 2A 9E C4 AF 32 A8 C4 CD E8 :8AC
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
CD00 CC 3A A2 C4 E6 80 CA 15 CD 2A A3 C4 CD E8 CC 3A :9CA
CD10 A7 C4 E6 80 C0 3E 01 32 A8 C4 C9 00 CD 8E CC 2A :888
CD20 9E C4 AF 32 A8 C4 CD E8 CC 3A A2 C4 E6 B0 C8 3E :8A4
CD30 01 32 A8 C4 C9 00 00 00 2A AB C4 3A CB C4 47 3A :64B
CD40 CA C4 A0 4F 06 00 09 4E 09 23 3E 3F A4 67 22 AB :55B
CD50 C4 79 32 A9 C4 3E 03 A1 32 AA C4 C9 AF 32 4C C5 :819
CD60 CD 38 CD 32 9A C4 CD F4 CC 3A A8 C4 A7 C8 3A 9A :9D8
CD70 C4 3C E6 03 32 9A C4 3A 4C C5 3C 32 4C C5 FE 04 :745
CD80 C2 66 CD 32 9A C4 C9 00 AF 32 4C C5 CD 38 CD 32 :844
CD90 9A C4 CD 1C CD 3A A8 C4 A7 C8 3A 9A C4 3C E6 03 :8E6
CDA0 32 9A C4 3A 4C C5 3C 32 4C C5 FE 04 C2 92 CD 32 :7AF
CDB0 9A C4 C9 00 3E 1B 32 AE C4 3E 02 32 AD C4 2A AD :6DE
CDC0 C4 22 90 C4 22 98 C4 CD 00 CC 3A 94 C4 A7 C2 DA :926
CDD0 CD CD 5C CD 3A 9A C4 FE 04 C0 3A AD C4 C6 02 32 :8C2
CDE0 AD C4 FE 1D FA BE CD 3A AE C4 D6 02 32 AE C4 F2 :A2B
CDF0 B9 CD E1 C9 CD 5C CD 3A 9A C4 FE 04 C8 AF 2A A0 :A01
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
CE00 C4 77 2A A5 C4 77 2A A3 C4 22 98 C4 C3 F4 CD 00 :8D8
CE10 21 02 1B 22 98 C4 CD F4 CD CD B4 CD C3 16 CE 00 :83F
CE20 01 8F 03 11 01 C1 21 00 C1 3E 80 77 ED B0 C9 00 :5E3
CE30 47 3A 9A C4 A7 C8 3D CB 20 C3 34 CE AF 32 9A C4 :87A
CE40 CD 1C CD 3A A8 C4 CD 30 CE 2A 9B C4 7E B0 77 3A :88F
CE50 9A C4 3C 32 9A C4 FE 04 C2 40 CE C9 3E 1C 32 AE :7FF
CE60 C4 3E 01 32 AD C4 2A AD C4 22 90 C4 22 98 C4 CD :802
CE70 00 CC 3A 94 C4 A7 CC 3C CE 3A AD C4 3C 32 AD C4 :865
CE80 FE 1D C2 66 CE 3A AE C4 3D 32 AE C4 A7 C2 61 CE :936
CE90 C9 00 00 00 CD 20 CE CD 10 CE 21 0E 1C 22 90 C4 :5F0
CEA0 CD 00 CC 2A 92 C4 AF 77 CD 90 D1 C9 AF F5 CD 64 :A0B
CEB0 06 21 41 00 19 E5 21 3F 00 19 D1 01 40 00 ED B8 :496
CEC0 F1 3C FE 15 C2 AD CE C9 AF F5 CD 64 06 D5 EB 23 :A04
CED0 23 D1 01 40 00 ED B0 F1 3C FE 15 C2 C9 CE C9 00 :834
CEE0 3E 13 F5 CD 64 06 D5 21 78 00 19 EB E1 01 42 00 :613
CEF0 ED B0 F1 3D F2 E2 CE C9 AF F5 CD 64 06 21 78 00 :9AA
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
CF00 19 01 42 00 ED B0 F1 3C FE 14 C2 F9 CE C9 00 00 :78A
CF10 3A CA C4 D6 05 32 4D C5 3A CB C4 D6 03 32 4E C5 :7CE
CF20 AF 32 4F C5 F5 2A 4D C5 22 90 C4 CD 00 CC AF F5 :8D9
CF30 3A 4F C5 CD 64 06 3A 96 C4 12 13 3A 97 C4 12 3A :61F
CF40 4F C5 3C 32 4F C5 F1 3C FE 03 C2 2F CF 3A 4E C5 :7D1
CF50 3C 32 4E C5 F1 3C FE 07 C2 24 CF C9 3A CA C4 C6 :8BF
CF60 05 32 4D C5 3A CB C4 D6 03 32 4E C5 AF 32 4F C5 :725
CF70 F5 2A 4D C5 22 90 C4 CD 00 CC AF F5 3A 4F C5 CD :8FF
CF80 64 06 21 40 00 19 ED 5B 96 C4 73 23 72 3A 4F C5 :5DC
CF90 3C 32 4F C5 F1 3C FE 03 C2 7B CF 3A 4E C5 3C 32 :777
CFA0 4E C5 F1 3C FE 07 C2 70 CF C9 F5 D5 2A 4D C5 22 :937
CFB0 90 C4 CD 00 CC D1 AF F5 3A 96 C4 12 13 3A 97 C4 :8B0
CFC0 12 13 F1 3C FE 03 C2 B7 CF 3A 4D C5 3C 32 4D C5 :767
CFD0 F1 3C FE 0B C2 AA CF C9 3A CA C4 D6 05 32 4D C5 :921
CFE0 3A CB C4 D6 03 32 4E C5 AF CD 64 06 CD 06 D8 C9 :841
CFF0 3A CA C4 D6 05 32 4D C5 3A CB C4 C6 03 32 4E C5 :7BE
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D000 3E 14 CD 64 06 CD 06 D8 C9 00 00 00 CD AC CE CD :711
D010 10 CF C9 CD C8 CE CD 5C CF C9 CD E0 CE CD D8 CF :BBB
D020 C9 CD F8 CE CD F0 CF C9 2A B3 C4 7E 4F 32 B5 C4 :ACA
D030 CB 3F 47 3A B0 C4 90 32 B7 C4 81 00 32 B9 C4 23 :78F
D040 7E 4F 32 B6 C4 CB 3F 47 3A B1 C4 90 32 B8 C4 81 :838
D050 3D 32 BA C4 23 C9 00 00 3A B7 C4 FE 42 D2 7B D0 :7EB
D060 3A B9 C4 A7 FA 7B D0 3A B8 C4 FE 15 D2 7B D0 3A :9C3
D070 BA C4 A7 FA 7B D0 AF 32 BF C4 C9 3E 01 32 BF C4 :98B
D080 E1 C9 00 00 06 00 3A B7 C4 A7 F2 92 D0 ED 44 4F :7E0
D090 09 AF 32 BB C4 3A B5 C4 4F ED 43 C6 C4 3A B8 C4 :8DB
D0A0 A7 F2 A9 D0 09 3C C2 A4 D0 32 BD C4 22 C0 C4 C9 :9AF
D0B0 3A B9 C4 FE 43 DA BA D0 3E 42 32 BC C4 3A BA C4 :946
D0C0 FE 15 DA C7 D0 3E 14 32 BE C4 C9 00 3A B9 C4 32 :83C
D0D0 BC C4 3A BA C4 32 BE C4 C9 00 00 00 3A BB C4 47 :7B5
D0E0 3A BC C4 90 4F 06 00 ED 43 C4 C4 3A B5 C4 4F ED :846
D0F0 43 C6 C4 3A BD C4 32 C8 C4 CD 64 06 3A BB C4 6F :8A5
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D100 26 00 19 22 C2 C4 C9 00 3A BD C4 47 3A BE C4 90 :6FE
D110 2A C0 C4 ED 5B C2 C4 ED 4B C4 C4 ED B0 2A C2 C4 :A89
D120 01 78 00 09 22 C2 C4 2A C0 C4 ED 4B C6 C4 09 22 :6C5
D130 C0 C4 3D F2 10 D1 C9 00 3A BD C4 47 3A BE C4 90 :8AB
D140 21 B8 CA ED 5B C2 C4 ED 4B C4 C4 ED B0 2A C2 C4 :A7E
D150 01 78 00 09 22 C2 C4 3D F2 40 D1 C9 CD 28 D0 CD :7C5
D160 58 D0 CD 84 D0 CD B0 D0 CD DC D0 3A B2 C4 A7 C2 :B28
D170 38 D1 CD 08 D1 C9 00 00 CD 28 D0 CD 84 D0 CD CC :8F7
D180 D0 CD DC D0 3A B2 C4 A7 C2 38 D1 CD 08 D1 C9 00 :9DA
D190 3E 38 F5 CD F8 D7 3C CB 27 32 98 C4 CD F8 D7 3C :99B
D1A0 CB 27 3C 32 99 C4 C3 D8 D7 2A A0 C4 AF 77 F1 3D :911
D1B0 C2 92 D1 C9 00 00 00 00 00 00 00 00 00 00 00 00 :2EE
D1C0 21 B6 C6 19 6E 26 C5 E5 DD E1 21 D6 C4 19 7E C9 :8CD
D1D0 21 BE C6 19 6E 26 C4 E5 FD E1 21 D0 C4 19 7E C9 :8EE
D1E0 3A 50 C5 06 00 4F C6 C8 21 E4 CB 09 CD 0C D8 09 :6C5
D1F0 6E 67 22 B3 C4 CD F4 D8 00 21 D4 CB 09 6E 01 06 :745
```

```
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D200 00 11 B5 C4 ED B0 CD 18 D8 C3 F0 DD CD 88 CD 3A :9D0
D210 9A C4 E1 FE 04 DD 77 07 F0 E5 C9 00 DD 7E 06 A7 :942
D220 C0 DD 66 01 DD 6E 00 22 98 C4 CD 0C D2 3A A2 C4 :818
D230 E6 40 CA 3B D2 3E FF DD 77 06 C9 2A 9B C4 7E E6 :94A
D240 DF 77 2A A0 C4 3E 20 B6 77 2A 9E C4 DD 75 00 DD :82A
D250 74 01 C9 00 DD 7E 06 A7 C8 3D DD 77 06 C9 00 00 :66E
D260 98 C4 DD 7E 07 32 9A C4 CD 8E CC 3A A2 C4 E6 40 :93B
D270 C0 DD 77 06 C9 00 00 00 3A CA C4 DD BE 00 C0 3A :740
D280 CB C4 DD BE 01 C0 3E 02 32 50 C5 DD 7E 07 32 9A :7A0
```

```
D290 C4 AF 32 B2 C4 CD E0 D1 3E 01 32 CF C4 32 51 C5 :8E5
D2A0 C9 00 00 00 11 07 00 D5 CD C0 D1 A7 C2 B8 D2 00 :707
D2B0 00 00 CD 1C D2 CD 78 D2 D1 3A 51 C5 A7 C0 1D F2 :869
D2C0 A7 D2 C9 00 3A CC C4 32 9A C4 CD 60 CC 11 07 00 :7AD
D2D0 D5 C5 CD C0 D1 C1 A7 C2 E9 D2 DD 7E 03 90 DD 77 :B1F
D2E0 03 DD 7E 02 91 91 DD 77 02 D1 1D F2 D0 D2 C9 00 :823
D2F0 3A CC C4 32 9A C4 CD 60 CC 11 05 00 D5 C5 CD D0 :8A0

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D300 D1 C1 A7 CA 15 D3 FD 7E 03 90 FD 77 03 FD 7E 02 :8ED
D310 91 91 FD 77 02 D1 1D F2 FC D2 C9 00 DD 7E 07 E6 :957
D320 03 4F 06 00 3A 56 C5 CB 21 E6 01 81 4F 21 AE C6 :5E5
D330 09 6E 26 C6 22 B3 C4 C9 11 07 00 D5 CD C0 D1 A7 :7B7
D340 C2 83 D3 CD 1C D3 DD 66 03 DD 6E 02 22 B0 C4 3E :83B
D350 01 32 B2 C4 CD 5C D1 DD 7E 06 A7 00 C2 7C D3 CD :889
D360 EC D8 CD 60 CC DD 7E 02 81 81 DD 77 02 00 00 32 :7A4
D370 B0 C4 DD 7E 03 80 DD 77 03 32 B1 C4 AF 32 B2 C4 :8A7
D380 CD 5C D1 D1 1D F2 3B D3 C9 00 00 00 ED 4B CA C4 :877
D390 7C 90 47 7D 91 4F CB 21 79 CB 21 81 C6 21 6F 78 :750
D3A0 CB 20 80 C6 0A 67 C9 00 06 03 9F F9 9F F9 9F F9 :83C
D3B0 9F F9 9F F9 9F F9 9F F9 9F F9 9F F9 06 03 A5 A5 :AE3
D3C0 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 A5 :A50
D3D0 11 05 00 D5 CD D0 D1 A7 CA 01 D4 E5 FD 66 03 FD :8E7
D3E0 6E 02 22 B0 C4 21 A8 D3 22 B3 C4 AF 32 B2 C4 CD :85F
D3F0 5C D1 E1 7E 3D 77 C2 01 D4 3E 01 32 B2 C4 CD 18 :7A3

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D400 D5 D1 1D F2 D3 D3 C9 00 3A CC C4 CB 27 47 3A 58 :8B9
D410 C5 E6 01 80 4F 06 00 21 B4 C7 09 6E 26 C7 22 B3 :656
D420 C4 C9 00 00 21 0B 1F 0A 21 09 23 0A 21 0A 00 00 :264
D430 21 0B 20 0A 21 09 22 0A 3A CC C4 CB 27 4F 06 00 :3BD
D440 21 24 D4 09 7E 32 B0 C4 23 7E 32 B1 C4 C9 00 00 :657
D450 3E 01 32 B2 C4 CD 08 D4 CD 38 D4 CD 5C D1 21 21 :7A5
D460 0A 22 B0 C4 AF 32 B2 C4 CD 5C D1 C9 1A 13 21 0C :714
D470 3A CC C4 4F 06 00 21 6C D4 09 6E 26 D0 E9 00 00 :5D6
D480 3A CC C4 FE 04 F0 CD 70 D4 CD C4 D2 CD F0 D2 C9 :B88
D490 3E 03 F5 CD 80 D4 CD 38 D3 00 00 00 CD B0 DC F1 :879
D4A0 3D C2 92 D4 C9 00 00 00 FE 04 E1 F0 E5 4F 06 00 :73B
D4B0 21 FC C6 09 7E C9 00 00 11 05 00 CD D0 D1 A7 C8 :726
D4C0 1D F2 BB D4 E1 C9 00 00 11 07 00 CD C0 D1 A7 C8 :82D
D4D0 1D F2 CB D4 CD 62 CB C9 00 00 00 00 3A CC C4 CD :808
D4E0 A8 D4 32 9A C4 2A CA C4 22 98 C4 CD 8E CC 3A A2 :945
D4F0 C4 E6 F0 C0 CD B8 D4 3E FF 77 CD B8 D8 77 FD 74 :BAC

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D500 05 FD 75 04 CD F2 E4 FD 74 01 FD 75 00 CD 8C D3 :92E
D510 FD 74 03 FD 75 02 C9 00 FD 66 05 FD 6E 04 AF 77 :7AE
D520 CD C1 E4 C9 00 00 00 00 3E 01 32 50 C5 AF 32 B2 :654
D530 C4 3A CC C4 32 9A C4 CD 54 D7 11 07 00 D5 CD C0 :890
D540 D1 A7 C2 7A D5 3A 9E C4 DD BE 00 C2 7A D5 3A 9F :9AA
D550 C4 DD BE 01 C2 7A D5 3E 01 77 D1 CD AC D8 3E 08 :88F
D560 93 07 07 07 07 47 21 42 C5 7E 80 27 77 23 3E 00 :41B
D570 8E 27 77 23 3E 00 8E 27 77 C9 D1 1D F2 3D D5 C9 :73D
D580 AF 32 50 C5 32 B2 C4 3A CC C4 32 9A C4 CD E0 D1 :976
D590 CD 4C D7 06 10 21 42 C5 7E 90 27 77 23 7E DE 00 :659
D5A0 27 77 23 7E DE 00 27 77 D0 3E 01 32 CF C4 32 51 :612
D5B0 C5 AF 77 2B 77 2B 77 C9 00 00 00 00 3A CC C4 FE :6C0
D5C0 04 C8 32 9A C4 2A CA C4 22 98 C4 CD 8E CC 3A A2 :895
D5D0 C4 47 E6 C0 C0 78 E6 20 28 05 CD 28 D5 18 03 CD :7CE
D5E0 80 D5 3E 01 32 B2 C4 CD 3C DE 3E 04 32 CC C4 C9 :7F0
D5F0 AF 32 CD C4 32 CE C4 3E 04 32 CC C4 3A CF C4 A7 :8AE

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D600 C0 DB 00 47 3E 04 A0 3E 02 CA 29 D6 3E 10 A0 3E :5F9
D610 03 CA 29 D6 3E 40 A0 3E 01 CA 29 D6 DB 01 4F 3E :65B
D620 01 A1 3E 00 CA 29 D6 3E 04 32 CC C4 DB 08 E6 40 :6B6
D630 C2 38 D6 3E 01 32 CE C4 DB 05 E6 01 C2 44 D6 3E :7B4
D640 01 32 CD C4 DB 09 E6 01 C0 E1 E1 C9 2A CA C4 22 :8B4
D650 98 C4 3A CC C4 32 9A C4 CD 8E CC 3A A2 C4 E6 F0 :A53
D660 CA 69 D6 3E 04 32 CC C4 C9 2A 9E C4 22 CA C4 C9 :8DB
D670 3A CD C4 A7 C4 DC D4 3A CE C4 A7 C4 BC D5 3A CC :AB4
D680 C4 FE 04 FC 4C D6 C9 00 AF 32 52 C5 3A CA C4 FE :96B
D690 0E C0 3A CB C4 A7 C0 3E 01 32 52 C5 32 CF C4 C9 :814
D6A0 CD D0 D3 CD A4 D2 CD F0 D5 3A CF C4 A7 F5 CC 70 :BEA
D6B0 D6 F1 CC 78 D8 CD 2C E4 CD 88 D7 CD 90 D7 3A CF :B29
D6C0 C4 A7 CA A0 D6 C9 00 00 06 00 21 FA C7 09 6E 26 :6F9
D6D0 C7 22 B3 C4 AF 32 B2 C4 CD 78 D1 C9 E6 0F 4F CD :9A7
D6E0 C8 D6 C9 00 D6 F0 0F 0F 0F 0F 4F CD C8 D6 C9 00 :7EC
D6F0 A7 CA FF D6 3A 4C C5 47 C5 3E 00 D3 40 0D C2 FD :8BA

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D700 D6 AF D3 40 C1 0D C2 05 D7 C9 00 00 00 00 00 00 :5CD
D710 4E 23 7E 23 F5 CD F8 D6 F1 3D C2 14 D7 C9 00 00 :846
D720 7E FE FF C8 CD 10 D7 C3 20 D7 00 00 00 00 00 00 :6B1
D730 80 20 20 30 80 80 20 30 FF FF FF FF 20 20 FE 20 :79A
D740 FF FF FF FF 21 30 D7 C3 20 D7 00 00 21 3C D7 C3 :8D5
D750 20 D7 00 00 CD E0 D1 CD 44 D7 C9 00 FE FE 10 FE :930
D760 20 FE 30 FE FF FF 21 5C D7 C3 20 D7 21 06 46 22 :7E7
D770 63 EA 21 44 C5 E5 CD BD 5E E1 2B E5 CD BD 5E E1 :9FE
D780 2B E5 CD BD 5E E1 C9 00 CD D0 D3 CD 6C D7 C9 00 :9EB
D790 CD C0 D8 21 0C 46 22 63 EA 21 53 C5 CD BD 5E 3A :7A2
D7A0 56 C5 C6 01 32 56 C5 D0 3A 53 C5 D6 01 27 32 53 :6D4
D7B0 C5 A7 C0 3E 01 32 55 C5 32 CF C4 C9 00 00 00 00 :645
```

```
D7C0 00 00 00 00 00 00 00 00 00 00 00 00 00 98 D7 CD :23C
D7D0 B4 D7 C9 21 E0 C5 09 C9 AF 32 9A C4 CD 8E CC 3A :98C
D7E0 A2 C4 A7 C2 A9 D1 3A 9A C4 3C 32 9A C4 FE 04 FA :9A9
D7F0 DC D7 C3 93 D1 C9 C9 C9 CD 38 CD 3A A9 C4 E6 0F :AA3

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D800 FE 0D F8 E6 0C C9 AF CD AA CF C9 00 6E 26 CB F5 :9D0
D810 3A 9A C4 4F F1 C9 00 00 2A B3 C4 C9 AF 32 D0 C4 :880
D820 01 0D 00 11 D1 C4 21 D0 C4 ED B0 C9 AF 32 CF C4 :843
D830 32 51 C5 32 52 C5 32 55 C5 32 56 C5 00 00 00 00 :52A
D840 00 21 0E 1C 22 CA C4 CD B8 DC 11 07 00 D5 CD C0 :6D6
D850 D1 3E 0E 6F DD 77 00 3E 01 67 DD 77 01 CD E4 D8 :764
D860 DD 75 02 DD 74 03 D1 1D F2 4D D8 CD 1C DD C9 00 :83C
D870 00 00 00 00 00 00 00 00 3A 58 C5 3C 32 58 C5 CD :3AF
D880 90 D4 C9 00 00 00 00 00 CD 2C D8 CD A0 D6 C9 00 :70A
D890 3A 9A C4 01 00 00 FE 04 E1 F0 E5 C9 3E 04 DD 77 :7B0
D8A0 07 3A 58 C5 E6 0F C0 CD 1C D2 C9 FF 2A A0 C4 AF :8D3
D8B0 77 CD C4 DD C9 FF FF FF 3E 40 2A A0 C4 C9 FF FF :B7E
D8C0 11 07 00 D5 CD C0 D1 A7 C2 DD D8 CD 54 D2 DD 66 :99F
D8D0 01 DD 6E 00 CD 8C D3 DD 75 02 DD 74 03 D1 1D F2 :800
D8E0 C3 D8 C9 00 CD 8C D3 AF DD 77 06 C9 DD 7E 07 32 :8F6
D8F0 9A C4 C9 00 3A 9A C4 4F 3A 50 C5 FE 02 C8 3A 9A :7F9

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D900 C4 CD A8 D4 4F C9 00 00 3E 01 32 55 C5 32 CF C4 :775
D910 C9 00 00 00 CD 88 D7 CD AC D7 C9 FF FF FF FF FF :B09
D920 AF 32 5F EA 01 19 50 CD 3A 09 3E 01 32 61 EA 3E :59E
D930 F8 32 5B EA 01 FF 00 CD F7 08 3E 0C CD 57 02 C9 :774
D940 3E FF 32 5F EA 01 19 24 CD 3A 09 C9 3E 03 32 C9 :60B
D950 C4 32 60 C5 32 61 C5 AF 32 42 C5 32 43 C5 32 44 :70B
D960 C5 32 66 C5 32 64 C5 CD 02 16 3A 76 EA 32 AB C4 :79D
D970 AF 32 53 C5 C9 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A 2A :490
D980 2A 2A 2A 2A 2A FF 60 6C 78 84 90 9C A8 B4 60 60 :6E1
D990 3E 01 F5 4F 06 00 21 F8 DA 09 6E 26 DB 2B E5 2B :62F
D9A0 D1 01 0C 00 ED B8 F1 3C FE 0A C2 92 D9 C9 00 00 :7AE
D9B0 AF F5 4F 06 00 21 F8 DA 09 6E 26 DB E5 23 D1 01 :73E
D9C0 0C 00 ED B0 F1 3C FE 09 C2 B1 D9 C9 3E 08 F5 4F :87C
D9D0 06 00 21 F8 DA 09 6E 26 DB E5 01 0D 00 A7 ED 42 :63A
D9E0 D1 ED B0 F1 3D C2 CE D9 C9 00 00 00 AF F5 4F 06 :8C7
D9F0 00 21 F8 DA 09 6E 26 DB E5 01 0D 00 09 D1 ED B0 :6D5

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
DA00 F1 3C FE 08 C2 ED D9 C9 21 02 DB 22 5A C5 3A CA :8C7
DA10 C4 D6 06 32 4D C5 3A CB C4 D6 04 32 4E C5 AF F5 :870
DA20 2A 4D C5 22 90 C4 CD 00 CC 3A 95 C4 2A 5A C5 77 :79E
DA30 01 0D 00 09 22 5A C5 3A 4E C5 3C 32 4E C5 F1 3C :553
DA40 FE 09 C2 1F DA C9 00 00 21 0E DB 22 5A C5 3A CA :6DA
DA50 C4 C6 06 32 4D C5 3A CB C4 D6 04 32 4E C5 AF C3 :82E
DA60 1F DA 00 00 21 02 DB 22 5A C5 3A CA C4 D6 06 32 :60E
DA70 4D C5 3A CB C4 D6 04 32 4E C5 AF F5 2A 4D C5 22 :7FC
DA80 90 C4 CD 00 CC 3A 95 C4 2A 5A C5 77 23 22 5A C5 :7A4
DA90 3A 4D C5 3C 32 4D C5 F1 3C FE 0D C2 7B DA C9 00 :7E4
DAA0 21 6A DB 22 5A C5 3A CA C4 D6 06 32 4D C5 3A CB :794
DAB0 C4 C6 04 32 4E C5 AF C3 7B DA 00 00 CD 90 D9 CD :89D
DAC0 EC DC C9 CD B0 D9 CD F8 DC C9 CD CC D9 CD 04 DD :C71
DAD0 C9 CD EC D9 CD 10 DD C9 CA C3 D1 BC 00 00 00 00 :8F8
DAE0 00 3A CC C4 FE 04 F0 4F 06 00 21 D8 DA 09 6E 26 :681
DAF0 DA 3E 01 32 59 C5 E9 00 02 0F 1C 29 36 43 50 5D :4CE

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
DB00 6A 77 58 B8 58 B8 58 B8 B8 B8 58 B8 58 B8 58 58 :851
DB10 B8 58 B8 58 B8 58 58 58 58 B8 58 B8 58 B8 58 B8 :820
DB20 58 B8 58 B8 58 B8 58 B8 58 58 58 58 58 58 58 58 :700
DB30 58 58 58 58 58 58 B8 B8 B8 B8 B8 B8 F8 B8 B8 B8 :980
DB40 B8 B8 58 B8 B8 B8 B8 B8 B8 B8 B8 B8 B8 B8 B8 B8 :B20
DB50 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 :F80
DB60 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 F8 :F80
DB70 F8 F8 F8 F8 F8 F8 F8 03 10 10 0F 10 10 10 10 10 :832
DB80 0C 0C 0C 0C 0C 0C 0C 0C 08 07 08 07 07 07 07 07 :09A
DB90 CD E0 DC 4F 06 00 21 78 DB 09 6E 26 DB 22 5D C5 :70E
DBA0 21 80 DB 09 7E 21 88 DB 09 4E 47 C9 A7 CA B3 DB :7ED
DBB0 FE 03 C0 CD DC DA 3E 04 32 CC C4 3E 03 C9 00 00 :752
DBC0 C5 CD 5B 06 2A 5D C5 3A 60 C5 CB 27 32 5F C5 C1 :7A7
DBD0 C5 AF 12 05 7E 13 12 13 23 3A 5F C5 12 13 C6 06 :4B3
DBE0 32 5F C5 7E 12 05 C2 D7 DB C1 C9 00 CD 98 E3 32 :863
DBF0 5C C5 AF F5 CD C0 DB 3A 5C C5 3D E6 03 C2 0C DC :958

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
DC00 2A 5D C5 11 0D 00 19 22 5D C5 3E 03 32 5C C5 F1 :54C
DC10 3C FE 15 C3 B0 E3 C9 00 3E 03 32 61 C5 3A 60 C5 :766
DC20 3C FE 04 C2 28 DC 3E 01 CD AC DB 32 60 C5 CD 90 :84B
DC30 DB CD EC DB C9 00 00 00 3E 03 32 61 C5 3A 60 C5 :730
DC40 3D CD AC DB 32 60 C5 CD 90 DB CD EC DB C9 00 00 :97D
DC50 3E 03 32 60 C5 3A 61 C5 3C FE 04 C2 60 DC 3E 01 :673
DC60 CD AC DB 32 61 C5 CD 90 DB CD EC DB C9 00 00 00 :941
DC70 3E 03 32 60 C5 3A 61 C5 3D CD AC DB 32 61 C5 CD :7AE
DC80 90 DB CD EC DB C9 00 00 21 03 03 22 60 C5 CD 90 :793
DC90 DB CD EC DB C9 00 00 00 50 38 70 18 88 88 88 88 :768
DCA0 3A CC C4 4F 06 00 21 98 DC 09 6E 26 DC E9 00 00 :616
DCB0 CD 50 D4 CD 40 DC C9 00 CD 1C D8 21 03 03 22 60 :76D
DCC0 C5 3E 04 32 CC C4 3E F8 32 02 DB 01 75 00 11 03 :598
DCD0 DB 21 02 DB ED B0 C9 00 3E 01 32 61 EA C9 00 00 :6C4
DCE0 3E F8 32 3C DB 3A CC C4 C9 00 00 00 CD 08 DA 3E :6FF
```

BUG FIRE!／プログラム・リスト

```
DCF0 58 32 3D DB C9 00 00 00 CD 48 DA 3E 58 32 3B DB :638

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
DD00 C9 00 00 00 CD 64 DA 3E 58 32 49 DB C9 00 00 00 :589
DD10 CD A0 DA 3E 58 32 2F DB C9 FF FF FF 3A CA C4 D6 :A7D
DD20 05 32 4D C5 3A CB C4 D6 03 32 4E C5 3E 0C CD 57 :69E
DD30 02 AF 32 4F C5 32 B2 C4 F5 2A 4D C5 22 90 C4 CD :813
DD40 8C D3 22 B0 C4 CD 24 E4 3A 94 C4 E6 40 CA 58 DD :981
DD50 2A 92 C4 AF 77 C3 69 DD 3A 94 C4 E6 80 CA 69 DD :9B7
DD60 21 BC D3 22 B3 C4 CD 5C D1 3A 4D C5 3C 32 4D C5 :80F
DD70 F1 3C FE 0B C2 38 DD 3A CA C4 D6 05 32 4D C5 3A :82E
DD80 4E C5 3C 32 4E C5 3A 4F C5 3C 32 4F C5 FE 07 3E :6A7
DD90 00 C2 38 DD 21 0E 10 22 CA C4 3A CB C4 3C 32 CB :6C8
DDA0 C4 CD EC D9 CD A0 DA 3A CB C4 FE 1C C2 9A DD CD :B86
DDB0 88 DC CD 88 E4 C9 FF FF FF FF FF FF 00 00 00 00 :A60
DDC0 CD 1C DD C9 D5 CD 1C D3 DD 66 03 DD 6E 02 22 B0 :885
DDD0 C4 3E 01 32 B2 C4 CD 5C D1 D1 CD 53 E4 C9 00 00 :843
DDE0 3E 80 D3 51 CD D8 DC C9 2F 3D 49 3B 49 3B 2F 3D :70C
DDF0 21 E8 DD 3A 50 C5 FE 02 C2 FE DD 21 EC DD 3A 9A :990

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
DE00 C4 4F 06 00 09 6E 26 DB 3E D8 77 21 3C DB 77 CD :69A
DE10 48 DE 2A B3 C4 C3 7B D1 21 E8 DD 3A 50 C5 FE 02 :90B
DE20 C2 26 DE 21 EC DD 3A 9A C4 4F 06 00 09 6E 26 DB :715
DE30 3E 58 77 21 3C DB 77 CD 48 DE C9 FF CD E0 D1 CD :9C2
DE40 18 DE C9 00 00 00 00 00 21 10 DB 22 5D C5 01 07 :417
DE50 0C CD EC DB C9 00 00 00 01 18 01 ED 43 5D EA 7E :678
DE60 FE FF C8 23 E5 CD 57 02 E1 C3 5F DE C1 AB B3 B6 :AAB
DE70 DD CE B3 BA B8 FF C0 DE B2 20 FF 20 D2 DD FF D2 :BDE
DE80 B2 DB 20 C9 20 DA CD DE D9 20 20 FF D3 C1 BC DE :A61
DE90 B6 DD 20 FF CC DF DA B0 D4 B0 20 C6 DD BD DE B3 :B7C
DEA0 20 FF 53 43 4F 52 45 20 20 FF 48 49 47 48 20 53 :56D
DEB0 43 4F 52 45 20 FF 46 20 49 20 47 20 48 20 54 20 :45A
DEC0 21 21 21 21 21 21 21 FF 01 19 24 CD 3A 09 21 01 :356
DED0 0C 22 63 EA 21 6C DE CD 58 DE 21 04 05 22 63 EA :682
DEE0 21 76 DE CD 58 DE 21 64 C5 CD BD 5E 21 7B DE CD :8F1
DEF0 58 DE 21 07 05 22 63 EA 21 7F DE CD 58 DE 21 65 :6D9

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
DF00 C5 CD BD 5E 21 0A 05 22 63 EA 21 8C DE CD 58 DE :7DA
DF10 21 53 C5 CD BD 5E 21 0D 05 22 63 EA 21 94 DE CD :723
DF20 58 DE 3A C9 C4 00 F6 30 CD 57 02 21 10 05 22 63 :604
DF30 EA 21 A2 DE CD 58 DE 21 44 C5 CD BD 5E 21 43 C5 :8C9
DF40 CD BD 5E 21 42 C5 CD BD 5E 21 13 05 22 63 EA 21 :6C1
DF50 AA DE CD 58 DE 21 48 C5 CD BD 5E 21 47 C5 CD BD :958
DF60 5E 21 46 C5 CD BD 5E 21 16 05 22 63 EA 21 B6 DE :6D2
DF70 CD 58 DE C9 00 00 00 00 60 56 52 48 44 40 36 32 :508
DF80 28 24 20 16 12 08 04 01 3C 38 34 30 2C 28 24 20 :211
DF90 1C 18 14 10 0C 08 04 01 3A 66 C5 E6 0F 4F 06 00 :320
DFA0 21 78 DF 09 7E 32 65 C5 21 88 DF 09 7E 32 91 D1 :6FE
DFB0 CD C8 DE C9 CD 20 D9 CD 4C D9 3E 01 32 52 C5 00 :87C
DFC0 00 00 3A 64 C5 3C 27 32 64 C5 3A 66 C5 3C CD 59 :5E8
DFD0 E0 3E 99 32 53 C5 C3 F9 DF 3A C9 C4 3D 32 C9 C4 :95F
DFE0 C8 CD C8 DE 3E 06 F5 01 FF FF 0B 78 B1 A7 C2 EA :9FA
DFF0 DF F1 3D C2 E6 DF C3 FF DF CD 98 DF CD 94 CE CD :C75

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E000 20 D9 CD 14 E0 DB 09 E6 01 C8 3A 52 C5 A7 C2 A5 :8AC
E010 E0 C3 94 E0 CD 88 D8 11 07 00 D5 CD C0 D1 DD 66 :9D2
E020 01 DD 6E 00 22 90 C4 CD 00 CC AF 2A 92 C4 77 D1 :7D2
E030 1D F2 1A E0 11 05 00 D5 CD D0 D1 FD 66 01 FD 6E :831
E040 00 22 90 C4 CD 00 CC 2A 92 C4 AF 77 D1 1D F2 37 :7CC
E050 E0 C9 CD B4 DF CD 40 D9 C9 32 66 C5 3A 53 C5 47 :9AE
E060 3A 42 C5 80 27 32 42 C5 3A 43 C5 CE 00 27 32 43 :5CD
E070 C5 3A 44 C5 CE 00 27 32 44 C5 C9 00 3E 00 D3 40 :652
E080 C5 0B 78 B1 A7 C2 81 E0 AF D3 40 C1 0B 78 B1 A7 :921
E090 C2 8C E0 C9 3E 20 F5 01 00 02 CD 7C E0 F1 3D C2 :866
E0A0 96 E0 C3 D9 DF 3E 0A F5 01 00 10 CD 7C E0 F1 3D :896
E0B0 C2 A7 E0 C3 C2 DF 00 00 3A 44 C5 47 3A 48 C5 90 :80E
E0C0 FA DC E0 C0 3A 43 C5 47 3A 47 C5 90 FA DC E0 C0 :A4B
E0D0 3A 42 C5 47 3A 46 C5 90 FA DC E0 C9 2A 42 C5 22 :82F
E0E0 46 C5 3A 44 C5 32 48 C5 21 D4 C7 CD 58 DE C9 00 :815
E0F0 CD 52 E0 CD C6 C9 C9 00 0C 20 2A 2A 2A 2A 2A 2A :64C

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E100 20 42 55 47 20 46 49 52 45 20 2A 2A 2A 2A 2A 2A :360
E110 0D 0A 0A 1C 1C 1C B7 B0 20 C9 20 BE C2 D2 B2 FF :6E8
E120 5B 58 5D A5 A5 A5 A5 BD C4 AF CA DF B0 FF FF :AD0
E130 3C 53 48 49 46 54 3E A5 A5 CA DD CF B0 FF FF FF :965
E140 5B 38 5D FF 5B 36 5D FF 5B 32 5D FF 5B 34 5D FF :7B0
E150 B5 C4 20 A6 20 B9 BC CF BD B6 3F 20 28 79 2F 6E :7B3
E160 29 FF FF FF 01 19 24 CD 3A 09 21 F8 E0 CD 58 DE :870
E170 21 06 06 22 63 EA 21 40 E1 CD 58 DE 21 0A 01 22 :52F
E180 63 EA 21 4C E1 CD 58 DE 21 0A 0B 22 63 EA 21 44 :6A8
E190 E1 CD 58 DE 21 0E 06 22 63 EA 21 48 E1 CD 58 DE :7D5
E1A0 21 07 07 CD F3 03 3E 1E 77 21 0A 04 CD F3 03 3E :4F5
E1B0 1D 77 21 0A 0A CD F3 03 3E 1C 77 21 0D 07 CD F3 :552
E1C0 03 3E 1F 77 21 08 14 22 63 EA 21 20 E1 CD 58 DE :5A8
E1D0 21 0C 14 22 63 EA 21 30 E1 CD 58 DE 21 10 01 22 :539
E1E0 63 EA 21 50 E1 CD 58 DE CD 75 0F 3A 69 EA FE 79 :8F7
E1F0 C2 FD E1 AF 32 FA D6 32 7D E0 C3 05 E2 3E 20 32 :91A

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E200 FA D6 32 7D E0 CD 10 E4 C9 00 00 00 0C 1C 1C 3C :669
E210 B9 DE B0 D1 20 C9 20 BE C2 D2 B2 3E 0D 0A 0A B9 :83D
E220 DE B0 D1 20 3C 20 42 20 55 20 47 20 20 46 20 49 :4E8
E230 20 52 20 45 20 3E 20 C9 20 BE C2 D2 B2 20 A6 20 :628
E240 B2 C0 BC CF BD A1 0D 0A 0A B1 C5 C0 20 CA 20 BE :87A
E250 B2 B9 DE DD BC DE B6 DD C5 B2 20 C6 20 D2 B2 DB :B2F
E260 20 A6 20 C0 DE AF BC AD C2 20 BC C5 B9 DA CA DE :A3A
E270 20 C5 D8 CF BE DD A1 0D 0A 0A BC B6 BC 20 BF C9 :8BF
E280 C0 D2 C6 CA 20 D2 B2 DB 20 C9 20 C5 B6 20 A6 20 :90B
E290 CA BC D8 CF DC D9 20 41 4C 49 45 4E 20 3D 20 3C :724
E2A0 42 55 47 3E 20 20 A6 20 BD CD DE C3 20 D4 AF C2 :7B2
E2B0 B9 C8 CA DE 20 C5 D8 CF BE DD A1 0D 0A 0A CA DE :9BA
E2C0 B8 DE 20 CA 20 B1 C5 C0 20 C9 20 B5 B2 C0 20 7B :8A1
E2D0 53 54 4F 50 50 45 52 7D 20 C6 B1 C0 D9 C4 0D 0D :6B8
E2E0 0A BC CA DE D7 B8 20 B3 BA DE B6 C5 B8 20 C5 D8 :A58
E2F0 CF BD A1 0D 0A 0A BF BA A6 20 CA DD CF B0 20 C3 :896

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E300 DE 20 C0 C0 B7 C2 CC DE BC C3 20 B8 C0 DE BB B2 :B03
E310 A1 0D 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 :0B8
E320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E330 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E340 00 00 00 00 00 0D 0A 0A DC B6 D8 CF BC C0 B6 3F :5CB
E350 20 28 79 2F 6E 29 FF FF DE BB B2 A1 29 0D 0A 0A :6BB
E360 DC B6 D8 CF BC C0 B6 3F 20 28 79 2F 6E 29 FF FF :92F
E370 21 0C E2 CD 58 DE CD 75 0F 3A 69 EA FE 79 C2 70 :899
E380 E3 CD 20 D9 C9 CD 64 E1 CD F0 E0 C9 FF FF FF FF :CE6
E390 CD 85 E3 C3 B8 E3 FF FF 3A 61 C5 F5 3A CC C4 FE :BAE
E3A0 04 F2 AC E3 3A FA D6 D3 40 F1 C9 00 F1 C9 00 00 :916
E3B0 C2 F3 DB AF D3 40 C9 00 21 D4 E3 CD 58 DE CD 75 :A38
E3C0 0F 3A 69 EA FE 20 CA 90 E3 C9 00 00 CD 90 E3 C3 :8C3
E3D0 66 5C 00 00 0D 0A 0A 20 20 20 20 52 45 50 4C 41 :2D7
E3E0 59 20 28 20 BD CD DF B0 BD 20 CA DE B0 20 29 FF :857
E3F0 0D 0A 0A 20 20 20 20 BE C2 D2 B2 20 B6 DE 20 B2 :62B

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E400 D8 CF BD B6 3F 28 79 2F 6E 29 FF 00 00 00 00 00 :5BF
E410 21 F0 E3 CD 58 DE CD 75 0F 3A 69 EA FE 79 C0 CD :9D9
E420 70 E3 C9 00 AF 32 94 C4 CD 00 CC C9 21 04 46 22 :744
E430 63 EA 21 48 E4 CD 58 DE 21 0A 46 22 63 EA 21 4E :6EC
E440 E4 CD 58 DE CD 88 D6 C9 53 43 4F 52 45 FF 54 49 :8F3
E450 4D 45 FF 06 00 11 07 00 CD C0 D1 80 47 1D F2 58 :63B
E460 E4 3E 08 90 F6 30 21 10 49 22 63 EA CD A6 E4 C5 :7E5
E470 CD C8 D4 C1 00 00 00 00 58 D5 21 0E 46 22 63 EA :63B
E480 21 A0 E4 CD 58 DE D1 C9 3E 38 21 10 49 22 63 EA :7A1
E490 CD 57 02 21 0E 46 22 63 EA 21 A0 E4 CD 58 DE C9 :77B
E4A0 41 4C 49 45 4E FF C5 CD 57 02 C1 C9 FF FF FF FF :9D9
E4B0 3A CC C4 FE 04 C8 21 EC DD 4F 06 00 09 CD F9 E4 :886
E4C0 C9 CD 5C D1 CD 28 CB C9 32 0E C1 CD 00 E5 C9 00 :8C8
E4D0 ED 44 C6 06 4F 06 00 C5 3A CB C4 FD 96 01 ED 44 :7A5
E4E0 C6 04 4F 06 00 21 F8 DA 09 6E 26 DB C1 09 3E 58 :5EA
E4F0 77 C9 CD B0 E4 2A 9E C4 C9 6E 26 DB 3E 98 77 C9 :97B

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E500 CD 2C E5 11 07 00 D5 CD C0 D1 3E 0E 6F DD 77 00 :738
E510 3E 01 67 DD 77 01 CD E4 D8 DD 75 02 DD 74 03 3E :76A
E520 04 DD 77 07 D1 1D F2 06 E5 C9 00 00 01 07 00 11 :50C
E530 D7 C4 21 D6 C4 AF 32 D6 C4 FF FF FF FF FF FF FF :CCA
```



チェック・サム　プログラム・リスト

```
0 CLEAR300,&HC000
10 DEFFNCH$(E,P)=STRING$(E-LEN(HEX$(P)),"0")+HEX$(P)
20 INPUT"Start add";S$:S=VAL("&H"+LEFT$(S$,LEN(S$)-1)+"0")
30 INPUT"End   add";E$:E=VAL("&H"+E$)
40 FOR A=S TO E STEP 16: SUM=0 : IF RIGHT$(HEX$(A),2)="00" OR A=S THEN GOSUB90
50 LPRINT FNCH$(4,A);" ";
60 FOR I=0 TO 15 : D=PEEK(I+A) :LPRINT FNCH$(2,D);" ";:SUM=SUM+D:NEXT
70 LPRINT":";FNCH$(3,SUM):NEXT
80 END
90 LPRINT:LPRINT"Add  ";:FOR I=0 TO 15:LPRINT"+";HEX$(I);" ";:NEXT:LPRINT" Sum":
RETURN
```



このゲームは，ゲームセンターで『KAMIKAZE』という名前がついているものとほとんど同じです．大型ＵＦＯから降りて来るエイリアンは，別にミサイルを発射することもなく，これは楽勝と思いきや，下に落ちると衝撃波(破片？）を出して，これに当たるとビーム砲はあっさりとやられてしまいます．

エイリアンがたくさん落ちてくるようになるともう避けようがなくてお手上げです．なかなか難しくて，ゲームセンターにあるものでは，まだ3,000点以上いったことがありません．

1 プログラムの入力

BASICのプログラムのうち，9000～9090はチェック・サムのプログラムです．BASICとマシン語のプログラムを入力して，ＲＵＮさせると64バイトごとのチェック・サムの値が次々と出てきますから，本の値と比べてください．チェック・サムがOKでしたら，BASICの行番号10と9000～9090を削除してください．

2 チェック・サム プログラム

８月号p.151に載せたチェック・サムのプログラムはかなり無駄な部分があります（ＰＣの **HEX$** 関数は２桁までしか使えないとばかり思っていたものですから！)．行番号200～220は今回のプログラムの9070のように１つにまとめられます．こうすれば260～290のサブルーチンもいらなくなり，かなり短くなります．

それから，240の**LPRINT**文はプリンタ出力をしないときは削除してください．さもないと，この行でプログラムが止まってしまいます．

3 遊び方

ビーム砲はテンキーの④で左へ，⑥で右へ移動します．ビームの発射は［スペース］キーです．

このプログラムはカラー，白黒どちらのモニタでも使えるようにしてあります．ＲＵＮすると，まずカラーＴＶかどうかを聞いてくるので，［Y］か［N］のキーを押してください．

画面右側にはSCOREとHi-SCOREが表示されます．画面中央の数字は局面数です．

右側から大型ＵＦＯが出てきて，中央にくるとエイリアンを降ろし始めます．エイリアンは溝に入り，ここへ５匹目が入ろうとすると１番下のエイリアンは押し出されて降下してきます．また１番端へ行ったエイリアンはすぐに降下してきます．

１番下へ落ちるとエイリアンは衝撃波を出しますから，その前に破壊するか逃げるかしてください．ときどき中央からＵＦＯが降りてきます．これは１番下へ着くと全面に衝撃波を出すため，必ずやられてしまいます．

ＵＦＯの両端に当てても爆発しませんし，時にはビームがすり抜けてしまうこともあります．ＵＦＯが出たらすぐに撃ちに行った方がいいでしょう．

50匹のエイリアンが降りると大型ＵＦＯは左へ移動して１面の終了です．また新たな大型ＵＦＯが右から出てきます．40面消し終わるとGIVE UPになります．

4 得点

溝の中に待機中のエイリアンは20点，降下しているエイリアンは40点です．UFOはMYSTERYで，100，200，300，400，500点のうちのどれか１つです．

5,000点を越えればビーム砲は１台追加されます．

5 プログラムの解説

1）マシン語

動きを必要とするところはすべてマシン語でプログラムしてあります．図１にメイン・プログラム，さらに図２～図６に，(１)～(５)のサブルーチンのフローチャートを示します．

ALIENの動きはⅠ～Ⅲの３つに分け，ALIEN Ⅰというのは中央の穴から１行下がったところまで，ALIENⅡはそ

写真1　大型UFOから次々とエイリアンが降りてくる.

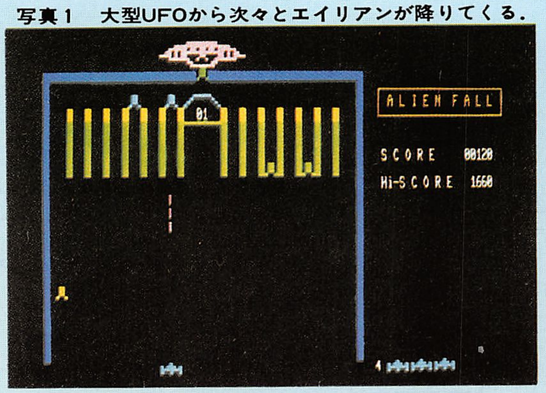


写真2　下に落ちたエイリアンは衝撃波をだす.



写真3　砲台が衝撃波に当たると, 簡単にやられてしまう.

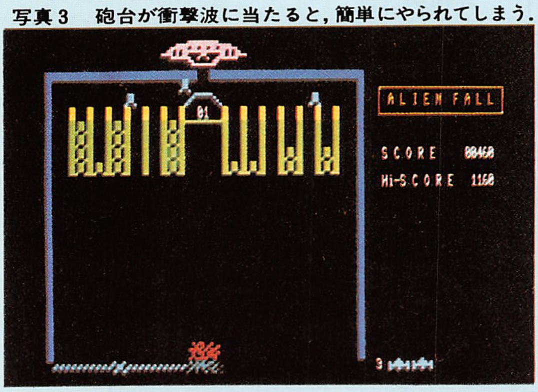


写真4　中央からUFOが降りてきたところ

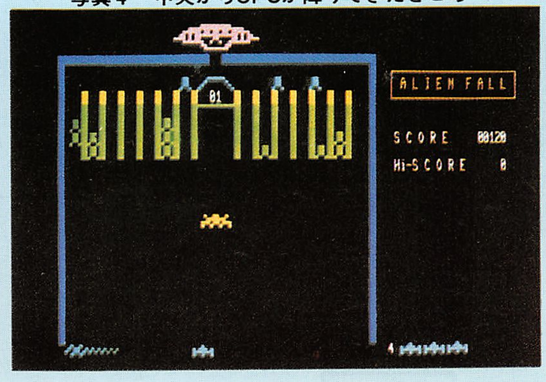


写真5　UFOのだす衝撃波には逃げることができない.

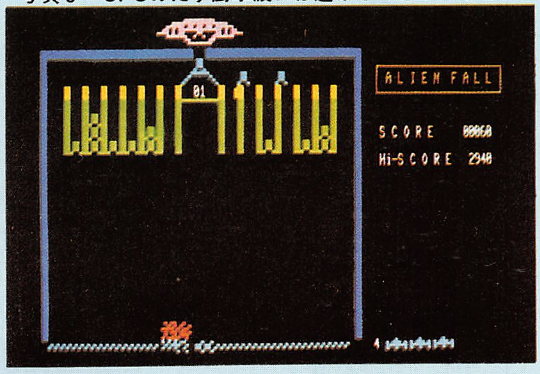
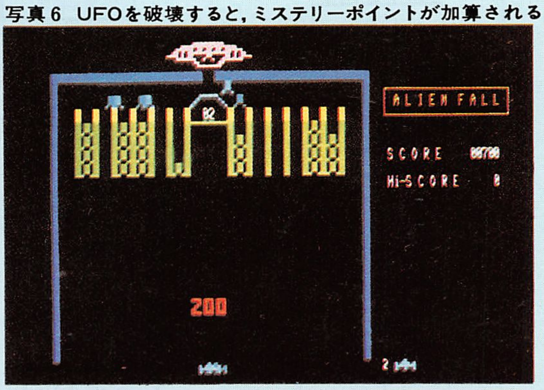


写真6　UFOを破壊すると, ミステリーポイントが加算される.



の続きで, 水平に右あるいは左の動きか, 溝に降下して待機に入るまで, ALIENⅢは降下している状態を示します. ALIENⅢのときにビームが当たれば40点が得られます.

ＵＦＯの発生はビームの発射数と乱数により, ビームを16発以上撃ったときに発生します.

マシン語プログラムの内容については**表1**に示します. ワーク・エリアを**表2**に, データ・エリアを**表3**に示します.

局面数が進むにつれて大型ＵＦＯからエイリアンの出てくるのが速くなってきます. 1, 2面あたりは遅くて易し過ぎる感じもしますが, しだいに速くなりますから安心してください.

2）BASIC

BASICプログラムの内容については**表4**に示します. 短いので説明の必要はないと思います. 420のキャラクタはキーボードでは[,]キー, 440のキャラクタは[N]キーによるものです.

6 変更点

変更できる項目について**表5**に示します. 数値は16進数です. 番号の頭に＊のついたものは, ０１, ０３, ０７,

図1　メイン・プログラム

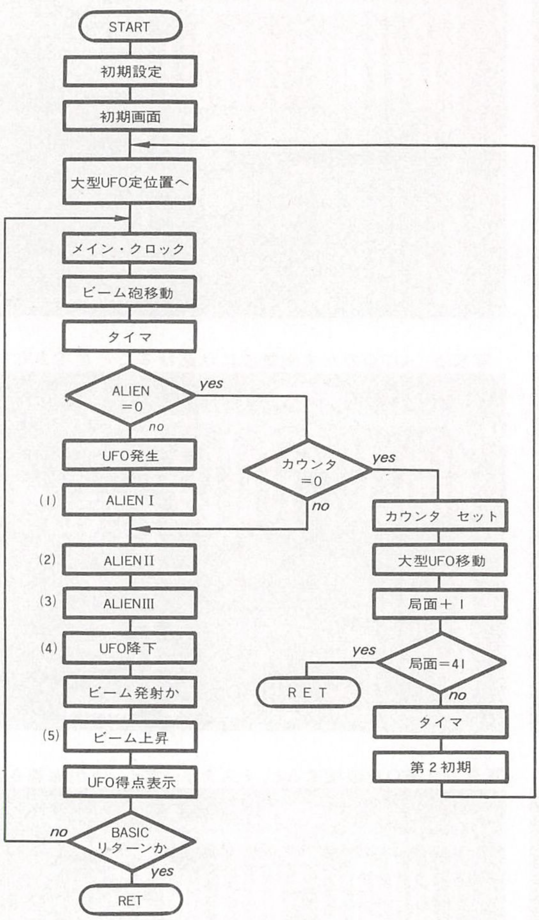


図3(a)　ALIEN II

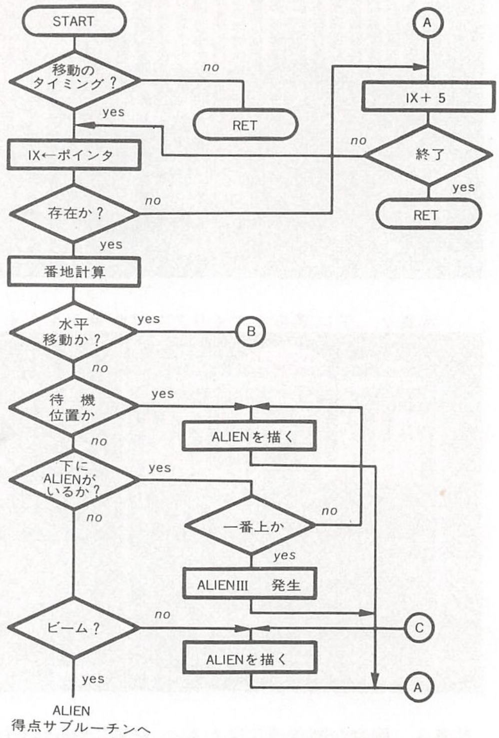


図3(b)　ALIEN II

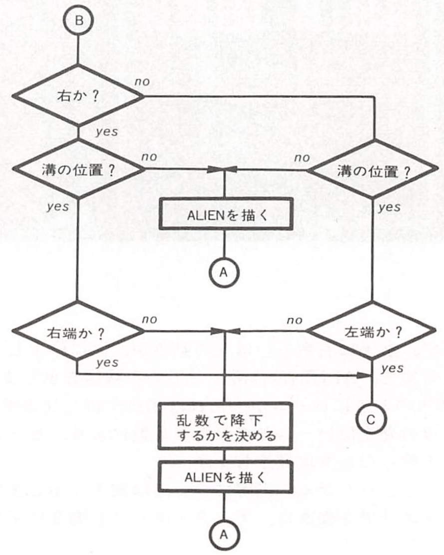


図2　ALIEN I

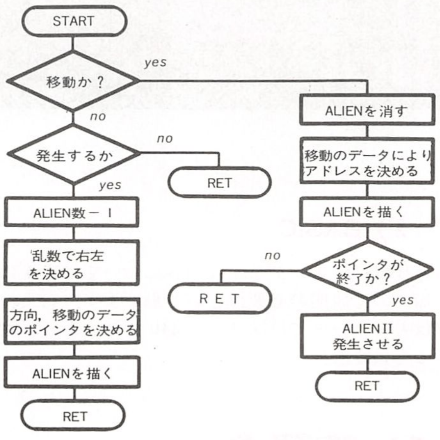


図４　ALIENⅢ

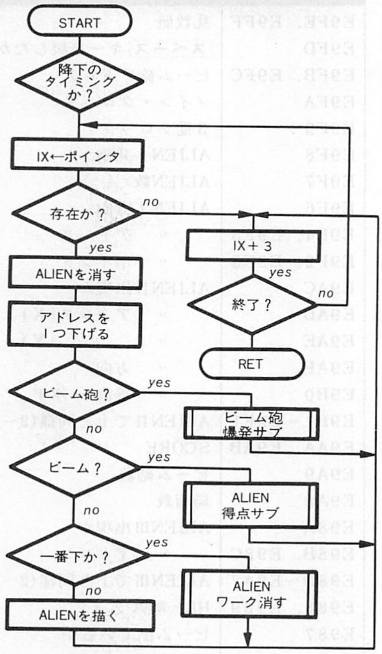


図５　ＵＦＯ降下

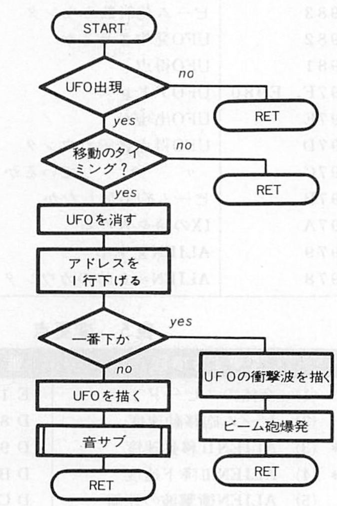
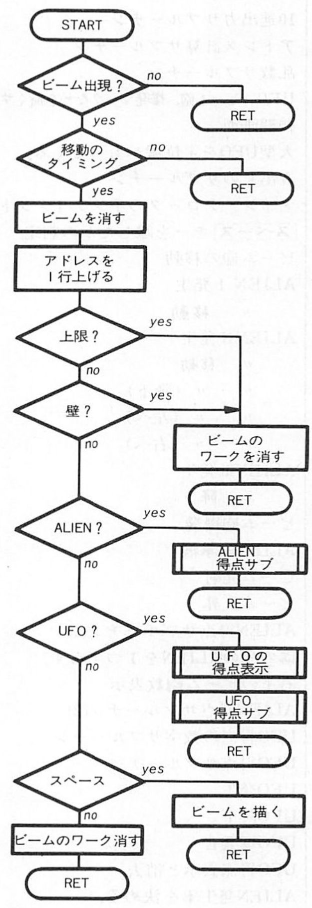


図６　ビーム上昇



０Ｆ，１Ｆ，３Ｆ……などの数値にしてください．

（２）のビーム砲移動速度は02以上の値にしてください．（５）の衝撃波の届く距離を小さくすればゲームは楽になります．（６）のＵＦＯ発生率はもう少し小さな数値にした方がいいかもしれません．ゲーム・センターのものは，かなり頻繁にＵＦＯが出てきます．

（７）のＵＦＯ降下速度は少し速すぎるようですが，この位の方がスリルがあっていいと思います．

７ あとがき

待望の夏休み．朝から晩までマイコンの前に座っていられるので，このゲームも４，５日で完成できると思っていたのですが，途中で遊びに行ったりしていたら調子が狂って結局１週間以上もかかってしまいました．ようやくアセンブラＲＯＭを手に入れたので，これからは少しは早くプログラムできるようになると思うのですが，使い方を覚えるまでが大変そうです．

このごろのゲーム・センターは回転が早くて，面白いゲ

表1　マシン語プログラム

アドレス	内　　　　容
D600～D64C	カラー化
D64D～D659	タイマ
D65A～D68C	10進出力サブルーチン
D68D～D6AF	アドレス計算サブルーチン
D6B0～D6C9	乱数サブルーチン
D6CA～D7A4	UFO，ビーム砲，爆発マークなどを描くサブルーチン
D7A5～D819	初期画面
D81A～D83F	大型UFOを定位置へもってくる．
D840～D883	音出しのサブルーチン
D884～D8A7	メイン・クロックのインクリメント
	スペース キーを離したかの判定
D8A8～D8E0	ビーム砲の移動
D8E1～D91F	ALIEN I 発生
D940～D979	〃　移動
D97A～D9B5	ALIEN II 発生
D9B6～D9F8	〃　移動
D9F9～DA5D	〃　〃（降下）
DA5E～DAF5	〃　〃（左へ）
DAF6～DB2C	〃　〃（右へ）
DB2D～DB61	ALIEN III 発生
DB62～DBE4	〃　降下
DBE5～DC95	ビーム砲爆発
DC96～DCFC	ALIEN衝撃波
DCFD～DD32	ビーム発射
DD33～DDAB	ビーム上昇
DDAC～DE6B	ALIEN得点サブルーチン(1)
DE6C～DE92	溝の中のALIENを1つずらす．
DE93～DEA7	右下のビーム砲数表示
DEA8～DECF	ALIEN得点サブルーチン(2)
DED0～DF48	UFO得点の数字サブルーチン
DF49～DF99	UFO得点サブルーチン
DF9A～DFCF	UFO発生
DFD0～E04F	UFO降下
E050～E067	UFO爆発音
E068～E08D	UFO得点表示と消去
E08E～E0B7	ALIEN発生率を決める．
E0B8～E12C	初期設定
E12D～E1BB	メイン・ルーチン
E1BC～E1DF	大型UFO左へ移動

表3　データ・エリア

アドレス	内　　　　容
D6FA～D718	大型UFOのデータ
D73A～D760	ビーム砲爆発マークのデータ
D84E～D85C	音出しのデータ
D920～D93F	ALIEN I 移動データ

表4　BASICプログラム

行番号	内　　　　容
140～150	初期設定
160～170	カラーか黒白かの選択
180，190	初期の画面を描く
200	マシン語ルーチンへ
210～220	ハイ・スコアの確認と表示
230～330	GAME OVER 処理
340～400	GIVE UP 処理
410～530	初期画面
9000～9090	チェック・サム

ームがあってプログラムし始めていると，次に見に行ったときにはもうなくなったりしていて困ってしまいます．

表2　ワーク・エリア

アドレス	内　　　　容	初期値
E9FE，E9FF	乱数値	
E9FD	スペース キーを戻したか	00
E9FB，E9FC	ビーム砲アドレス	FE59H
E9FA	メイン・クロック	
E9F9	3進クロック	00
E9F8	ALIEN I 発生	00
E9F7	ALIEN数	32H
E9F6	ALIEN I 方向	
E9F4，E9F5	〃　アドレス	
E9F2，E9F3	〃　ポインタ	
E9AC	ALIEN II 出現か	00
E9AD	〃　アドレス(X)	
E9AE	〃　〃　(Y)	
E9AF	〃　方向	
E9B0	〃　垂直 or 水平	
E9B1～E9F1	ALIEN II で上と同様(2～14)	
E9AA，E9AB	SCORE	0000
E9A9	ビーム砲数	04
E9A8	局面数	01
E98A	ALIEN III 出現か	00
E98B，E98C	〃　アドレス	
E98D～E9A7	ALIEN III で上と同様(2～10)	
E988，E989	HL一時スタック	
E987	ビーム出ているか	00
E985，E986	ビーム・アドレス	
E984	BASICリターンか	00
E983	ビーム発射数カウンタ	00
E982	UFO発生させるか	00
E981	UFO得点	00
E97F，E980	UFOアドレス	
E97E	UFO出現か	00
E97D	UFO得点表示カウンタ	
E97C	〃　〃　しているか	00
E97B	ビーム砲追加したか	01
E97A	IXの値を入れる	
E979	ALIEN発生率	0FH
E978	ALIEN＝0時のカウンタ	FFH

表5　変更点

変更する項目	番　地	現在の数値
(1)　全体のスピード	E13D	04
(2)　ビーム砲移動速度	D88D	03
＊(3)　ALIEN II 移動速度	D9BA	07
＊(4)　ALIEN III 降下速度	DB66	07
(5)　ALIEN衝撃波の距離	DCA4	0C
(6)　UFO発生率	DD25	11
＊(7)　UFO降下速度	DFDA	07
(8)　GIVE UPの局面数	E19C	29

いつもオリジナル・ゲームを作ろうとは思っているのですが，なかなか難しくて，結局ゲーム・センターのコピーということになってしまっています．

このゲームで筆者の最高点は21面で，25,760点でした．

●参考にしたもの

1) 阿蘇坊舞子："舞子のプログラム教室"，I/O，'80年7月号
2) ゲーム『平安京エイリアン』
3) ゲーム『KAMIKAZE』

ALIEN FALL BASICプログラム

```
10 GOTO 9000
100 '╒══════════════════════════════════════╕
110 '│  ALIEN FALL          VER 1.4         │
120 '│               1980年8月14日             │
130 '╘══════════════════════════════════════╛
140 CLEAR300,&HD5FF
150 CONSOLE0,25,0,1:WIDTH80,25:COLOR7,0,1:PRINTCHR$(12)
160 LOCATE5,12:PRINT"ｱﾅﾀ ﾉ TV ﾊ COLOR ﾃﾞｽｶ ?( Y or N )";
170 G$=INKEY$:IF G$="y"THEN C1=1:C2=2:C3=3:C4=4:C5=5:C6=6:C7=7 ELSE IF G$="n"THE
N C1=7:C2=7:C3=7:C4=7:C5=7:C6=7:C7=7 ELSEGOTO 170
180 GOSUB410:DEFUSR2=&HD7C0:AA=USR2(0):GOTO 270
190 GOSUB410
200 DEFUSR1=&HE12D:AA=USR1(0)
210 SC=PEEK(&HE9AB)*256+PEEK(&HE9AA)
220 IF SC>HS THEN HS=SC:LOCATE69,10:PRINTUSING"#####";HS
230 IF PEEK(&HE984)=5 THEN340
240 COLORC6:LOCATE10,9:PRINT"  ══════════════════════════════  "
250 LOCATE12,10:PRINT"**   G A M E   O V E R   **"
260 LOCATE12,11:PRINT"═════════════════════════════"
270 COLORC4:LOCATE10,13:PRINT"K E Y  F U N C T I O N (ﾃﾝｷｰ)"
280 COLORC7:LOCATE22,15:PRINT"4 <═══> 6"
290 LOCATE10,17:PRINT"B E A M  ······  S P A C E  K e y"
300 COLORC5:LOCATE7,20:PRINT"5000 point ｦ ｺｴﾙﾄ ﾋﾞｰﾑﾎｳ 1 ﾀﾞｲ ﾌｴﾏｽ｡"
310 COLORC7:LOCATE5,23:PRINT"H i t  R E T U R N  k e y  t o  S t a r t";
320 IF INPUT$(1)<>CHR$(13)THEN320 ELSE GOTO190
330 END
340 LINE(1,3)-(50,9)," ",BF
350 COLORC3:LOCATE12,7:PRINT" ══════════════════════════════ ";:BEEP
360 LOCATE14,8:PRINT"**   G I V E   U P   **";:BEEP
370 LOCATE14,9:PRINT"══════════════════════════";:BEEP
380 COLORC6:LOCATE16,13:PRINT"ALIEN ﾊ ｻﾘﾏｼﾀ｡"
390 FOR I=1TO2000:NEXT:GOTO 270
400 END
410 COLORC4,0,1:PRINTCHR$(12)
420 LINE(0,3)-(0,21),"■",C1:LINE(51,3)-(51,21),"■",C1
430 LINE(0,88)-(1,95),PSET,C1,B:LINE(102,88)-(103,95),PSET,C1,B
440 LINE(0,2)-(24,2),"■",C1:LINE(27,2)-(51,2),"■",C1
450 FOR I=44TO58STEP14:LINE(I,20)-(I+1,23),PSET,C6,BF:NEXT
460 LINE(46,28)-(57,39),PRESET,C3,BF
470 IF C4=4 THEN DEFUSR0=&HD60B:AA=USR0(0)
480 COLORC6:LOCATE54,3:PRINT"┌─────────────────────┐"
490 LOCATE54,4:PRINT"│A L I E N  F A L L│"
500 LOCATE54,5:PRINT"└─────────────────────┘"
510 COLORC7:LOCATE55,8:PRINT"S C O R E      00000"
520 LOCATE55,10:PRINTUSING"Hi-S C O R E  #####";HS
530 LOCATE53,24:PRINT" 4";:LOCATE25,5:PRINT"01":RETURN
9000 CLEAR300,&HD5FF
9010 WIDTH40,25:CONSOLE0,25,0,0:PRINTCHR$(12)
9020 PRINT"ALIEN FALL  CHECK SUM"
9030 PRINT"START=D600":S=54784!:PRINT"END   =E1DF":E=57823!:T=64:PRINT
9040 X=0:R=S+T:IF R>=E THEN R=E+1
9050 FOR I=S TO R-1:Y=PEEK(I)
9060 X=X+Y:NEXT
9070 S$=HEX$(S):S=S+T:E$=HEX$(R-1):X$=HEX$(X)
9080 PRINT S$;" - ";E$;" = ";X$
9090 IF S>=E THEN END ELSE GOTO 9040
```

チェック・サム

```
ALIEN FALL  CHECK SUM

START=D600
END  =E1DF
BYTE =64

D600 - D63F = 1B9E
D640 - D67F = 161E
D680 - D6BF = 12B1
D6C0 - D6FF = 11D4
D700 - D73F = 197E
D740 - D77F = 1A4F
D780 - D7BF = 162F
D7C0 - D7FF = 1D9A
D800 - D83F = 1745
D840 - D87F = 1F75
D880 - D8BF = 2367
D8C0 - D8FF = 1F3A
D900 - D93F = 16A0
D940 - D97F = 1C4D
D980 - D9BF = 1512
D9C0 - D9FF = 1716
DA00 - DA3F = 1A80
DA40 - DA7F = 1906
DA80 - DABF = 225E
DAC0 - DAFF = 175A
DB00 - DB3F = 1EEB
DB40 - DB7F = 14FF
DB80 - DBBF = 187A
DBC0 - DBFF = 1C05
DC00 - DC3F = 2420
DC40 - DC7F = 193C
DC80 - DCBF = 1521
DCC0 - DCFF = 1B54
DD00 - DD3F = 1F8C
DD40 - DD7F = 1BAA
DD80 - DDBF = 191B
DDC0 - DDFF = 15EE
DE00 - DE3F = 1D82
DE40 - DE7F = 1942
DE80 - DEBF = 17F7
DEC0 - DEFF = 0F6E
DF00 - DF3F = 0D80
DF40 - DF7F = 1AE2
DF80 - DFBF = 1C0E
DFC0 - DFFF = 20AA
E000 - E03F = 1A2B
E040 - E07F = 1BC3
E080 - E0BF = 1507
E0C0 - E0FF = 1708
E100 - E13F = 1C3A
E140 - E17F = 2203
E180 - E1BF = 1AC8
E1C0 - E1DF = 0BB2
```

注）チェック・サムはI/O '80年8月号，9月号のチェック・サム プログラムを利用してください．

ALIEN FALLマシン語プログラム

```
ALIEN FALL DUMP LIST

START=D600
END  =E1DF

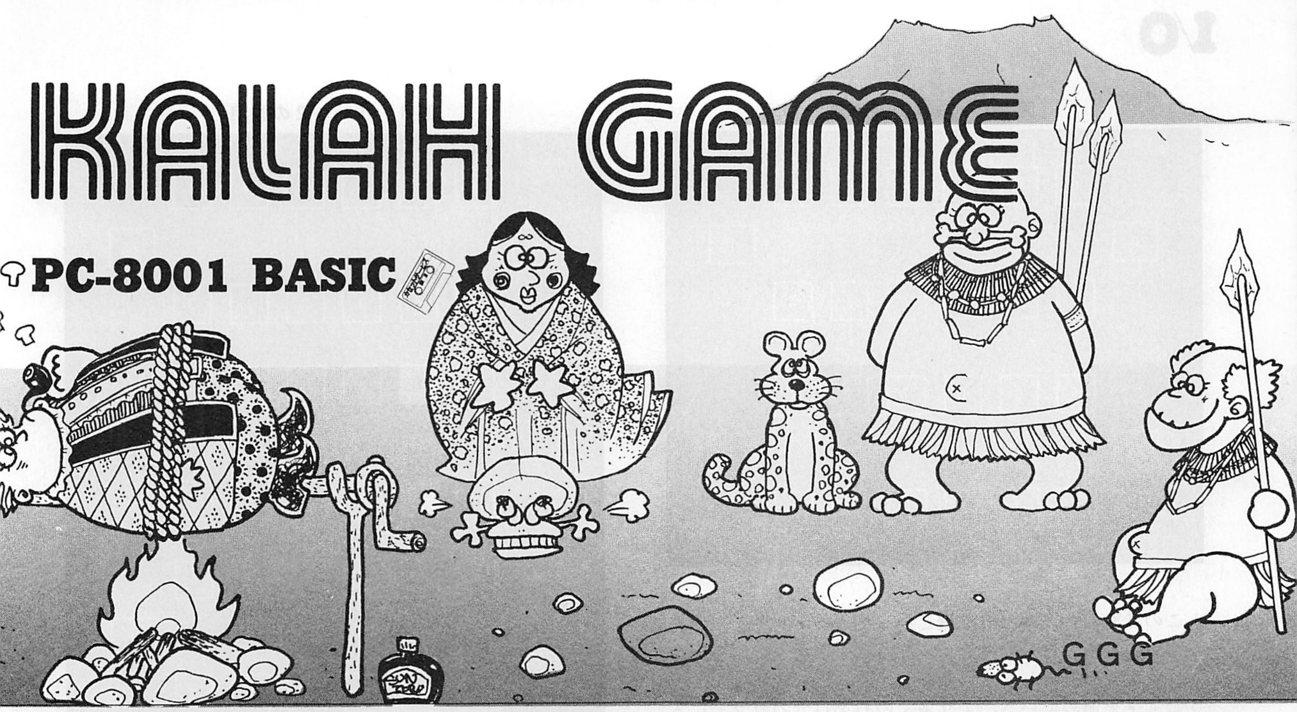
D600 : 11 78 00 77 19 10 FC C9 00 00 00 3E 78 32 53 F3
D610 : 32 CB F3 3E B8 32 BD F4 32 35 F5 3E D8 32 AD F5
D620 : 32 B5 F5 3E F8 32 B1 F5 21 05 F8 3E 78 06 04 CD
D630 : 00 D6 3E D8 06 05 CD 00 D6 3E 58 06 05 CD 00 D6
D640 : 3E B8 32 93 FE C9 00 00 00 00 00 00 00 16 05 1E
D650 : 00 1B 7A B3 20 FB C9 00 00 00 01 10 27 CD 79 D6
D660 : 01 E8 03 CD 79 D6 01 64 00 CD 79 D6 0E 0A CD 79
D670 : D6 7D C6 30 12 C9 00 00 00 C5 37 3F 3E 30 ED 42
D680 : 38 03 3C 18 F9 C1 09 12 13 C9 00 00 00 D5 21 00
D690 : 00 11 78 00 06 08 CB 0F 30 01 19 EB 29 EB 05 20
D6A0 : F5 11 00 F3 19 D1 16 00 19 C9 00 00 00 00 00 00
D6B0 : E5 D5 2A FE E9 54 5D 19 19 7D 84 67 11 81 00 19
D6C0 : 22 FE E9 7C D1 E1 C9 00 00 00 36 C0 23 36 65 23
D6D0 : 36 7A 23 36 A7 23 36 56 23 36 0C C9 00 00 00 00
D6E0 : 06 06 36 00 23 10 FB C9 00 00 00 36 4E 23 36 F6
D6F0 : 23 36 46 23 36 0E C9 00 00 00 11 31 32 32 63 8C
D700 : 9A A9 C8 36 23 23 13 11 00 C8 E4 E6 BE FF FB BF
D710 : FF EB 6E 4E 8C 00 00 00 00 06 0E 1A 77 23 13 10
D720 : FA C9 00 00 00 11 FA D6 CD 19 D7 D5 11 7A FF 19
D730 : D1 C3 19 D7 00 00 00 00 00 00 4A A0 31 68 28 A5
D740 : A5 61 41 0D 58 52 96 35 A4 41 53 D6 50 52 72 1A
D750 : A5 00 90 97 EB 69 D7 02 00 84 C4 80 04 08 00 00
D760 : 00 06 06 C3 1B D7 00 00 00 11 3A D7 2B CD 61 D7
D770 : D5 11 82 FF 19 D1 CD 61 D7 D5 11 82 FF 19 D1 C3
D780 : 61 D7 00 00 00 11 4C D7 C3 6C D7 00 00 00 00 00
D790 : 2B CD E0 D6 11 82 FF 19 CD E0 D6 11 82 FF 19 C3
D7A0 : E0 D6 00 00 00 06 07 36 FF 23 23 23 10 F9 23 23
D7B0 : 23 23 06 07 36 FF 23 23 23 10 F9 C9 00 00 00 00
D7C0 : 21 5D F5 CD A5 D7 21 D5 F5 CD A5 D7 21 4D F6 CD
D7D0 : A5 D7 21 C5 F6 CD A5 D7 21 3D F7 CD A5 D7 21 E8
D7E0 : F5 06 06 36 11 23 10 FB 21 F8 F4 36 48 23 36 12
D7F0 : 23 36 11 23 36 11 23 36 21 23 36 84 21 59 FE 22
D800 : FB E9 CD EB D6 21 79 FE 06 03 CD EB D6 23 10 FA
D810 : C9 00 00 00 00 00 00 00 00 00 21 B8 F3 0E 2D E5
D820 : CD 25 D7 11 78 00 19 36 00 E1 2B 16 05 CD 40 D8
D830 : 16 30 CD 4F D6 0D 20 E7 C9 00 00 00 00 00 00 00
D840 : 3E 20 D3 40 CD 4F D6 AF D3 40 C9 00 00 00 A1 C0
D850 : FF 38 90 F0 2F 70 BB CF 68 5F 05 07 00 E5 5E 23
D860 : 66 6D CD 6A D8 E1 C9 00 00 00 3E 20 D3 40 CD 7B
D870 : D8 AF D3 40 CD 7B D8 25 20 F0 C9 E5 2D 20 FD E1
D880 : C9 00 00 00 21 FA E9 34 3A F9 E9 3C FE 03 20 01
D890 : AF 32 F9 E9 3A FD E9 FE 01 C0 DB 09 FE FF C0 AF
D8A0 : 32 FD E9 C9 00 00 00 00 3A F9 E9 FE 01 C0 2A FB
D8B0 : E9 DB 00 FE EF CA CE D8 FE BF C0 00 00 00 7D FE
```

ALIEN FALLマシン語プログラム

```
D8C0 : 71 C8 36 00 23 22 FB E9 C3 EB D6 00 00 00 7D FE
D8D0 : 41 C8 2B 22 FB E9 CD EB D6 23 36 00 C9 00 00 00
D8E0 : 00 3A F8 E9 FE 00 C2 40 D9 CD 8E E0 A0 C0 00 3A
D8F0 : F7 E9 3D 32 F7 E9 CD B0 D6 E6 01 20 07 3E 01 21
D900 : 30 D9 18 04 AF 21 20 D9 32 F6 E9 22 F2 E9 21 0A
D910 : F4 22 F4 E9 36 78 23 36 87 3E 01 32 F8 E9 C9 00
D920 : 78 00 01 00 01 00 01 00 79 00 00 00 00 00 00 00
D930 : 78 00 FF FF FF FF FF FF 77 00 00 00 00 00 00 00
D940 : 3A FA E9 E6 07 C0 2A F4 E9 36 00 23 36 00 2B E5
D950 : 2A F2 E9 5E 23 56 23 22 F2 E9 E1 19 22 F4 E9 36
D960 : 78 23 36 87 3A F2 E9 FE 2A 20 05 21 1E 00 18 06
D970 : FE 3A C0 21 16 01 AF 32 F8 E9 DD 21 AC E9 06 10
D980 : DD 7E 00 FE 00 28 13 11 05 00 DD 19 10 F2 2A F4
D990 : E9 36 00 23 36 00 C9 00 00 00 DD 74 03 DD 75 01
D9A0 : DD 36 02 04 DD 36 00 01 DD 36 04 00 C9 00 00 00
D9B0 : 00 00 00 00 00 00 3A FA E9 E6 07 C0 DD 21 AC E9
D9C0 : 0E 0E DD 7E 00 FE 01 28 0C 11 05 00 DD 19 0D 20
D9D0 : F1 C9 00 00 00 DD 5E 01 DD 7E 02 CD 8D D6 36 00
D9E0 : 23 36 00 2B DD 7E 04 FE 00 20 0E DD 7E 03 FE 01
D9F0 : CA 5E DA C3 F6 DA 00 00 00 11 78 00 DD 34 02 DD
DA00 : 7E 02 FE 0A 20 0F 36 78 23 36 87 DD 36 00 00 C3
DA10 : C9 D9 00 00 00 19 7E FE 78 20 1C 11 88 FF 19 DD
DA20 : 35 02 DD 7E 02 FE 05 C2 06 DA 11 58 02 19 CD 2D
DA30 : DB C3 C9 D9 00 00 00 FE E0 20 09 CD B7 DE C3 C9
DA40 : D9 00 00 00 23 7E FE E0 28 F1 36 87 2B 36 78 C3
DA50 : C9 D9 00 00 00 00 00 00 00 00 00 00 00 00 2B DD
DA60 : 35 01 DD 7E 01 FE 15 20 0E CD B0 D6 E6 07 CA DE
DA70 : DA C3 E2 DA 00 00 00 FE 12 20 0E CD B0 D6 E6 07
DA80 : CA DE DA C3 E2 DA 00 00 00 FE 0F 20 0E CD B0 D6
DA90 : E6 03 CA DE DA C3 E2 DA 00 00 00 FE 0C 20 0E CD
DAA0 : B0 D6 E6 03 CA DE DA C3 E2 DA 00 00 00 FE 09 20
DAB0 : 0E CD B0 D6 E6 03 CA DE DA C3 E2 DA 00 00 00 FE
DAC0 : 06 20 0E CD B0 D6 E6 01 CA DE DA C3 E2 DA 00 00
DAD0 : 00 FE 03 20 0D CD 2D DB C3 C9 D9 00 00 00 DD 36
DAE0 : 04 01 36 78 23 36 87 C3 C9 D9 00 00 00 00 00 00
DAF0 : 00 00 00 00 00 00 23 DD 34 01 DD 7E 01 FE 1F CA
DB00 : 69 DA FE 22 CA 7B DA FE 25 CA 8D DA FE 28 CA 9F
DB10 : DA FE 2B CA B1 DA FE 2E CA C3 DA FE 31 CA D5 DA
DB20 : C3 E2 DA 00 00 00 00 00 00 00 00 00 00 DD 36 00
DB30 : 00 06 0A FD 21 8A E9 FD 7E 00 FE 00 28 0B 11 03
DB40 : 00 FD 19 10 F2 C9 00 00 00 FD 36 00 01 FD 75 01
DB50 : FD 74 02 36 78 23 36 87 C9 00 00 00 00 00 00 00
DB60 : 00 00 3A FA E9 E6 07 C0 06 0A DD 21 8A E9 DD 7E
DB70 : 00 FE 01 28 0B 11 03 00 DD 19 10 F2 C9 00 00 00
DB80 : DD 6E 01 DD 66 02 36 00 23 36 00 2B 11 78 00 19
DB90 : 0E 02 7E FE 4E 28 0C FE F6 28 08 FE 46 28 04 FE
DBA0 : 0E 20 09 CD E6 DB C3 75 DB 00 00 00 FE E0 20 09
DBB0 : CD C4 DE C3 75 DB 00 00 00 0D 28 04 23 C3 92 DB
DBC0 : 2B 7C FE FE 20 09 CD 96 DC C3 75 DB 00 00 00 DD
DBD0 : 75 01 DD 74 02 36 78 23 36 87 C3 75 DB 00 00 00
DBE0 : 00 00 00 00 00 C5 DD 36 00 00 2A FB E9 0E 05 E5
DBF0 : CD 85 D7 21 52 D8 CD 5D D8 E1 E5 CD 69 D7 21 58
DC00 : D8 CD 5D D8 21 59 D8 CD 5D D8 E1 E5 CD 90 D7 16
DC10 : 0C CD 4F D6 E1 0D C2 EF DB 2A 85 E9 36 00 AF 32
DC20 : 87 E9 21 51 FD CD 7A DC 21 84 FD CD 7A DC CD 88
DC30 : DC 21 77 FE 3A A9 E9 3D 32 A9 E9 FE 00 20 0C 36
DC40 : 30 3E 01 32 84 E9 C1 C9 00 00 00 E5 21 59 FE 22
DC50 : FB E9 CD EB D6 E1 F5 C6 30 77 23 23 F1 3D FE 00
DC60 : 20 02 C1 C9 47 CD EB D6 23 10 FA C1 C9 00 00 00
DC70 : 00 00 00 00 00 00 00 00 00 00 06 02 36 FF 11 78
DC80 : 00 19 10 F8 C9 00 00 00 06 4E 21 41 FE 36 00 23
DC90 : 10 FB C9 00 00 00 C5 DD 36 00 00 22 88 E9 36 69
DCA0 : 23 36 96 06 0C 23 7D FE 74 20 03 2B 18 0C 7E FE
DCB0 : 4E 20 07 C1 C3 E5 DB 00 00 00 36 24 E5 21 5A D8
DCC0 : CD 5D D8 2A 88 E9 2B 7D FE 41 20 03 23 18 0F 22
DCD0 : 88 E9 7E FE 0E 20 07 E1 C1 C3 E5 DB 00 00 36 24
DCE0 : E1 10 C2 16 05 CD 4F D6 06 34 CD 8A DC 2A FB E9
DCF0 : C1 C3 EB D6 00 00 00 00 00 00 00 00 00 3A FD E9
DD00 : FE 00 C0 3A 87 E9 FE 00 C0 DB 09 FE BF C0 3E 01
DD10 : 32 FD E9 32 87 E9 2A FB E9 23 22 85 E9 3A 83 E9
DD20 : 3C 32 83 E9 FE 11 C0 3E 01 32 82 E9 C9 00 00 00
DD30 : 00 00 00 3A 87 E9 FE 01 C0 3A FA E9 E6 01 C0 2A
DD40 : 85 E9 7E FE E0 20 02 36 00 11 88 FF 19 7E FE 85
DD50 : 28 08 FE 11 28 04 FE FF 20 08 AF 32 87 E9 C9 00
DD60 : 00 00 FE 78 28 05 FE 87 20 0B 2B AF 32 87 E9 C3
DD70 : AC DD 00 00 00 FE C0 28 04 FE 0C 20 03 C3 5A DD
DD80 : FE 65 28 0C FE 7A 28 08 FE A7 28 04 FE 56 20 0A
DD90 : AF 32 87 E9 CD 49 DF 00 00 00 FE 00 20 06 22 85
DDA0 : E9 36 E0 C9 AF 32 87 E9 C9 00 00 00 36 00 23 36
DDB0 : 00 2B 16 09 CD 40 D8 DD 21 8A E9 06 0A DD 7E 00
DDC0 : FE 01 28 0C 11 03 00 DD 19 10 F2 18 1B 00 00 00
DDD0 : DD 7E 01 BD 20 EE DD 7E 02 BC 20 E8 DD 36 00 00
DDE0 : 01 28 00 18 3C 00 00 00 EB DD 21 AC E9 06 0E DD
DDF0 : 7E 00 FE 01 28 11 D5 11 05 00 DD 19 D1 10 F0 CD
DE00 : 6C DE 18 1A 00 00 00 D5 DD 5E 01 DD 7E 02 CD 8D
DE10 : D6 D1 7D BB 20 E0 7C BA 20 DC DD 36 00 00 01 14
DE20 : 00 2A AA E9 09 22 AA E9 11 06 F7 CD 5A D6 2A AA
DE30 : E9 3A 7B E9 FE 00 C8 7C FE 13 20 07 A7 7D FE 88
DE40 : D8 18 04 A7 FE 13 D8 3A A9 E9 3C 32 A9 E9 21 77
DE50 : FE 00 CD 93 DE 06 0F 16 0D CD 40 D8 10 F9 AF 32
DE60 : 7B E9 C9 00 00 00 00 00 00 00 00 00 EB 06 04 E5
DE70 : 11 88 FF 19 7E FE 78 28 0A E1 36 00 23 36 00 C9
DE80 : 00 00 00 E1 36 78 23 36 87 2B 19 10 E2 C9 00 00
DE90 : 00 00 00 F5 C6 30 77 23 23 F1 3D 47 CD EB D6 23
DEA0 : 10 FA C9 00 00 00 00 00 36 00 AF 32 87 E9 16 09
DEB0 : C3 40 D8 00 00 00 00 CD A8 DE C5 CD 1A DE C1 C9
DEC0 : 00 00 00 00 CD A8 DE C5 CD DC DD C1 C9 00 00 00
DED0 : 36 00 19 36 00 23 36 01 11 88 FF 19 36 0F C9 00
DEE0 : 00 00 23 36 1F 23 36 0F 23 36 1F 23 36 0F 11 78
DEF0 : 00 19 36 01 2B 36 11 2B 36 01 2B 36 11 C9 00 00
DF00 : 00 36 59 19 36 11 23 36 01 11 88 FF 19 36 03 C9
DF10 : 00 00 00 36 51 19 36 11 23 36 01 11 88 FF 19 36
DF20 : 0F C9 00 00 00 36 8E 19 36 00 23 36 01 11 88 FF
DF30 : 19 36 0F C9 00 00 00 36 57 19 36 11 23 36 01 11
DF40 : 88 FF 19 36 0D C9 00 00 00 2A 7F E9 11 78 00 3A
DF50 : 81 E9 FE 32 20 05 CD D0 DE 18 1E FE 64 20 05 CD
DF60 : 01 DF 18 15 FE 96 20 05 CD 13 DF 18 0C FE C8 20
DF70 : 05 CD 25 DF 18 03 CD 37 DF CD E2 DE 3E 40 32 7D
DF80 : E9 3E 01 32 7C E9 3A 81 E9 6F 26 00 29 44 4D CD
DF90 : 50 E0 C3 21 DE 00 00 00 00 00 3A 81 E9 FE FA 20
DFA0 : 04 3E 32 18 02 C6 32 32 81 E9 3A 82 E9 FE 01 C0
DFB0 : CD B0 D6 E6 07 C0 AF 32 83 E9 32 82 E9 3E 01 32
DFC0 : 7E E9 21 60 F6 22 7F E9 C3 CA D6 00 00 00 00 00
DFD0 : 3A 7E E9 FE 01 C0 3A FA E9 E6 07 C0 2A 7F E9 E5
DFE0 : CD E0 D6 E1 11 78 00 19 7C FE FE 28 0F 22 7F E9
DFF0 : CD CA D6 21 59 D8 C3 5D D8 00 00 00 AF 32 7E E9
E000 : 22 88 E9 36 69 23 36 96 23 36 66 23 36 69 23 36
E010 : 96 06 16 23 7E FE 4E 20 0A DD 21 7A E9 C3 E5 DB
E020 : 00 00 00 36 24 E5 21 5A D8 CD 5D D8 2A 88 E9 2B
E030 : 22 88 E9 7E FE 0E 20 04 E1 C3 19 E0 36 24 E1 10
E040 : D2 2A FB E9 C3 19 E0 00 00 00 00 00 00 00 00 00
E050 : C5 AF 32 7E E9 06 0F 16 08 CD 40 D8 16 05 CD 4F
E060 : D6 10 F4 C1 C9 00 00 00 3A 7C E9 FE 01 C0 3A 7D
E070 : E9 3D 32 7D E9 FE 00 C0 32 7C E9 2A 7F E9 CD E0
E080 : D6 11 72 00 19 C3 E0 D6 00 00 00 00 00 00 3A A8
E090 : E9 FE 03 20 04 3E 07 18 0E FE 06 20 04 3E 03 18
E0A0 : 06 FE 0B 20 05 3E 01 32 79 E9 3A 79 E9 47 C3 B0
E0B0 : D6 00 00 00 00 00 00 00 AF 32 FD E9 32 F9 E9 32
E0C0 : AA E9 32 AB E9 32 87 E9 32 84 E9 32 81 E9 32 7C
E0D0 : E9 3E 04 32 A9 E9 3E 01 32 A8 E9 3E 0F 32 79 E9
E0E0 : 3E 01 32 7B E9 3E FF 32 78 E9 18 09 00 00 00 00
E0F0 : 00 00 00 00 00 06 0E 21 AC E9 36 00 23 23 23 23
E100 : 23 10 F7 06 0A 21 8A E9 36 00 23 23 23 10 F9 3E
E110 : 32 32 F7 E9 AF 32 F8 E9 AF 32 83 E9 32 82 E9 32
E120 : 7E E9 C9 00 00 00 00 00 00 00 00 00 00 CD B8 E0
E130 : CD C0 D7 CD 1A D8 CD 84 D8 CD A8 D8 16 04 CD 4F
E140 : D6 3A F7 E9 FE 00 28 0B CD 9A DF CD E1 D8 18 0E
E150 : 00 00 00 3A 78 E9 3D 32 78 E9 FE 00 28 1E CD B6
E160 : D9 CD 62 DB CD D0 DF CD FD DC CD 33 DD CD 68 E0
E170 : 3A 84 E9 FE 00 C0 C3 36 E1 00 00 00 3E A0 32 78
E180 : E9 CD BC E1 3A A8 E9 3C 32 A8 E9 F5 6F 26 00 11
E190 : 71 F5 CD 66 D6 21 71 F5 36 00 F1 FE 29 20 09 3E
E1A0 : 05 32 84 E9 C9 00 00 00 CD 0F E1 16 AF CD 4F D6
E1B0 : C3 33 E1 00 00 00 00 00 00 00 00 00 21 8B F3 0E
E1C0 : 12 CD 1F D8 21 01 F3 CD D3 E1 21 79 F3 C3 D3 E1
E1D0 : 00 00 00 06 0F 36 00 23 10 FB C9 00 00 00 00 00
```

KALAH GAME

PC-8001 BASIC



スペース・インベーダーの爆発的流行以来，ＴＶゲームは隆盛を極め，マイコン用のものもすでに数多く市販され，あるいは雑誌などでプログラムが発表されています．現在のところ，そのほとんどはアタマの体操というよりユビの運動に近い作品ですが，たまにはしっかり頭を使う遊びも良いのでは？　というわけで，ここに昔から知られている『カラー』というゲームを紹介します．

1 経緯と由来

ＴＫ大学では毎年×月になると×月祭という行事が催され，各学部学科によって様々な企画が行なわれるらしいのですが，Ｓ学部のＩＳ学科ではその日だけミニコン室がゲーム・センターに変容，各種コンピュータ・ゲームを発表して例年好調な客入り(？)を記録するそうです．

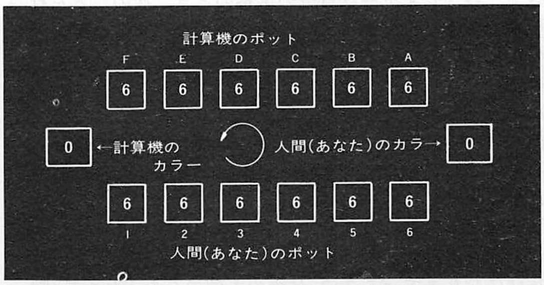
さて，昭和55年度×月祭の前日，ＩＳ学科ではすでに多くのユニークなゲームが製作され準備完了していたのですが，プログラムを作るのが大好きなＩ氏は夜になって『僕もゲームを作ろう』と言いだし，その晩徹夜してあるゲームを作りました．

×月祭が終わってから，のこのこ現われたＫ(本文担当)は勉強もそっちのけでしばらくこのゲームに熱中した後，『これは記事になるのでは』と考え，やはりプログラムを作るのが大好きなＡ氏に相談に行ったのです．

PC-8001＋高解像度カラーモニタを所有する金持ちのＡ氏は，『よろしい』と一言いって，〝C〞という言語で書かれていたオリジナル・プログラムをBASIC＋マシン語に変換し，カセットに入れてくれました(結局Ｋは何もしていない)．そのゲームというのが，実はこれから紹介する『カラー』です．

『カラー』は英語ではKALAHと綴(つづ)りますが，本来はアフリカの原住民が行なっていた石取りゲームのことです．比較的単純なルールの割には変化に富んでいるので，電子計算機が発明されてからは，人工知能ゲーム（計算機と人間が対戦するゲーム）の例題の１つとして良く知られてきました．

図１　盤面の説明



今回のプログラムも，人間と計算機とが対戦するという型式になっています(人間同士の対戦をシミュレートする機能はありません)．百円ＴＶゲームのような実時間性はありませんが，オセロやチェスと同様，じっくり考えてから手を選ぶという点で，知的なマイコン・ホビーストにふさわしいゲームと言うことができるでしょう．

なお，このゲーム自体についてはI/O別冊③『BASICゲーム徹底研究』でもすでに紹介されているので，お持ちの方はそちらもあわせて読まれるのが良いと思います．

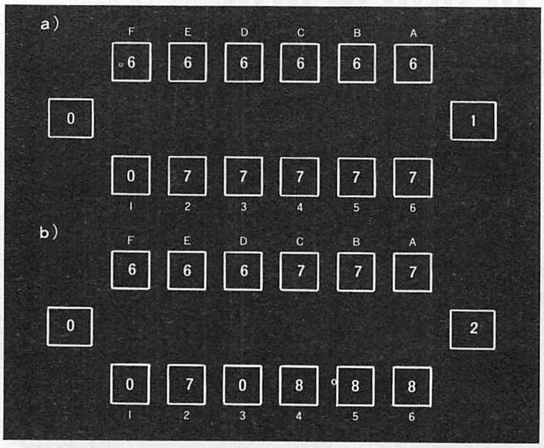
2 ルールと遊び方

ゲームは図１のとおり，14個の箱とその中にある合計72個の石からなります．箱には２種類あり，大きい箱は**カラー**（KALAH），小さい箱は**ポット**（POT）と呼ばれます．**図１**中，上に並んでいる６個のポットが計算機のポットで，下の６個が人間のポットです．

計算機のポットにはＡ～Ｆの記号が割り当てられており，人間のポットには１～６の数字が割り当てられています．ポットの左右にある大きい箱がカラーで，左が計算機のカラー，右が人間のカラーです．

これらの箱に石を入れるのですが，箱の中にある数字でその時点での箱の中の石の数を表わすことにします．ゲームの開始時には図のように各ポットに６個ずつの石が入っており，カラーには石が入っていません．

図2　ゲームの進め方



この状態から人間と計算機という2人のプレイヤーが交互に手を打ち，自分のカラーにできるだけたくさんの石を入れるように努力します．ただし，パスはできません．そうして手を進めていき，より多くの石を自分のカラーに入れたプレイヤーが勝ちます．

さて，各プレイヤーは自分の手番のとき，次のことを行ないます．まず自分のポットのうち空でない1つを選び，その中に入っている石を全部手に取ります（当然そのポットは空になる）．

次に，今，選んだポットのすぐ右にある箱から始めて，左回り（反時計回り）に順に1箱につき1個ずつ手にあった石を入れていきます．その際，相手のカラーだけは石を入れずに飛ばします．

たとえば，図2（a）の状態で人間の手番のとき，彼が3のポットを選んだとすると，石をばらまいた後は図2（b）の状態になります．ここで，最後の1個の石を入れた箱（図2では相手のポットC　が何であるかにより，次の3つのルールのうち，いずれか1つが適用されます．

●**ルール1**　自分のカラーの場合，再び同じプレイヤーの手番となります．

●**ルール2**　自分のポットで空のものの場合，そのポットのちょうど真向かいにある相手のポットが空でなければ，その2つのポットにある石を全部自分のカラーに移し，相手に手番を渡します．

たとえば，図3（a）の状態で計算機の手番のとき，彼がBのポットを選べば，最後の石を入れるのはEになりますが，Eはもともと空で，かつ真向かいにある人間のポット2が空ではないので，石をばらまいた後，図3（b）のように2つのポットの石を移動し，結局，図3（c）の状態になって人間の手番になるのです．

●**ルール3**　その他の場合，何もせず単に手番が移ります．

ゲームの進行中に，一方のプレイヤーのポットが全部空になることがしばしばありますが，その場合はポットに残っている石をすべて相手のプレイヤーのカラーに移し，ゲームを終了します．また，石の数は合計72個あり，途中で増えたり減ったりしないので，どちらかのカラーの石の数が37個以上になれば，そちらの勝ちということで，やはりゲームを終了します．

以上でルールの説明は終わりですが，実際にカラーを行なう場合は次の各点に注意するとよいでしょう．

❶序盤戦ではポットの中に石が多いので，**ルール2**をうまく使えば相手のポットの石をごっそり取ることができます．逆に相手に石を取られることにもなりやすいので注意が必要．この段階で相手にたくさんの石を取られると勝つ可能性は非常に少なくなります．

❷石をばらまく操作は13回で1回りします．そこで，自分のポットに石が13個たまったら，そのポットの石をばらまくようにすると，最後の石を入れる場所がちょうど元のポットになり，必ず**ルール2**が適用されて相手のポットの石を取ることができます（図4）．

❸中盤戦以後では，できるだけ自分の手番が多くなるよう，最後にカラーで終わるばらまき方を優先します．たとえば，図5（a）の状態で人間の手番のときは，4回連続し

図3　ルール2の説明

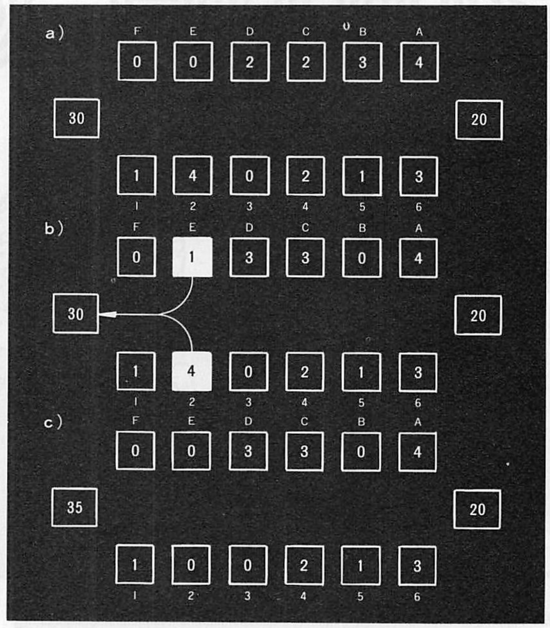


図4　石が13個あるときの戦略

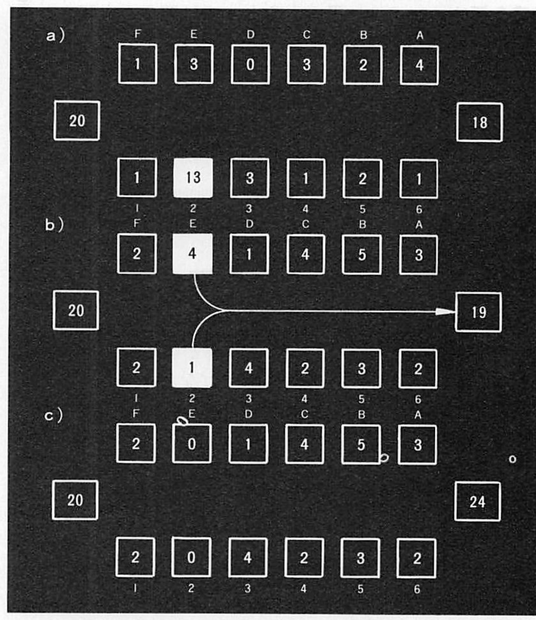
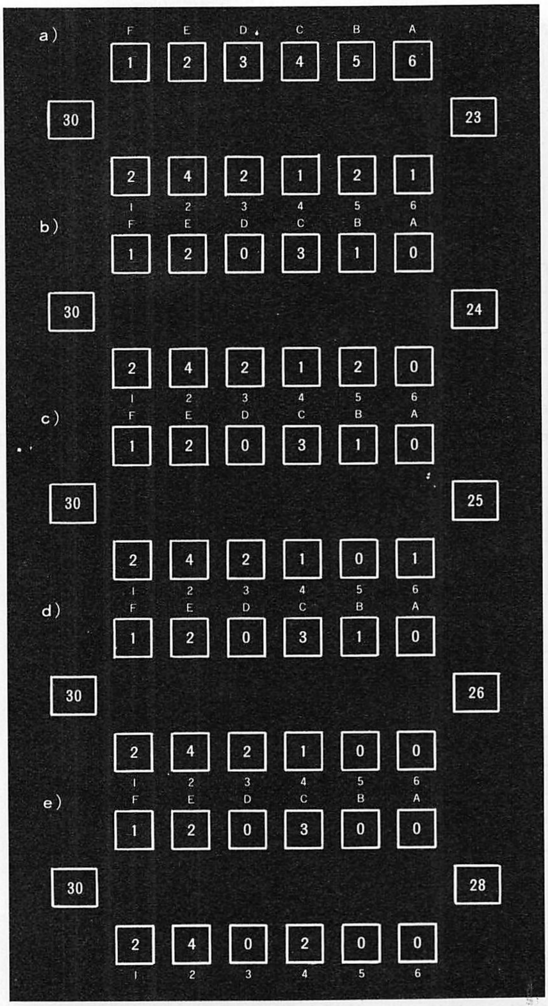


図5　連続して手が進められる例



て手を進めることができます．

❹中盤戦から終盤戦にかけては，相手のポットに石を回さないようにし相手のポットがすべて空になるのを待つのも1つの手です．特に今回のプログラムでは，計算機は自分のポットが空になることをまったく気にせずに手を打ってくるので，この❹の戦略はとりわけ重要になります．

❶〜❹以上のより精密な戦略は，実際に計算機と対戦して身につけるのが一番です．というわけで，具体的な操作手順の説明に移りましょう．

計算機をONにし，プログラムをロードしてRUNさせると，初めに『少し待て』といってくるので，少し待ちます．すると，画面に箱が表示され，まずレベルを聞いてくるので，0〜9程度の数を入力してやります(**写真1**)．入力の際は必ず最後に **return** キーを押してください．この注意は以下同様です．このレベルが大きいほど計算機は強くなります．なお，単に **return** を押すと，**6** が入力されたと解釈します．

次に，先手か後手かを聞いてきます(**写真2**)．先手でプレイしたければ **1** を，後手でしたければ **2** を入力してください．後はすぐゲームに入ります．

計算機の手番のときは計算機が勝手に手を進めますから，

写真1　初めに，相手の強さ(レベル)を入力する．

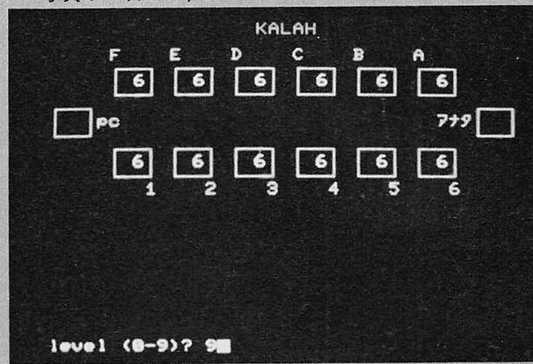


写真2　次に，先手か後手かを入力するとゲーム開始．

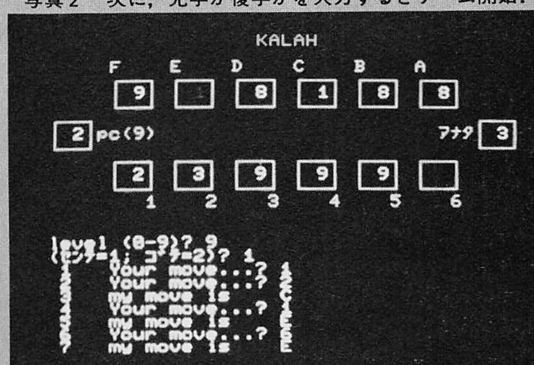


写真3　51手目で過半数の石を取って，挑戦者の勝ち．

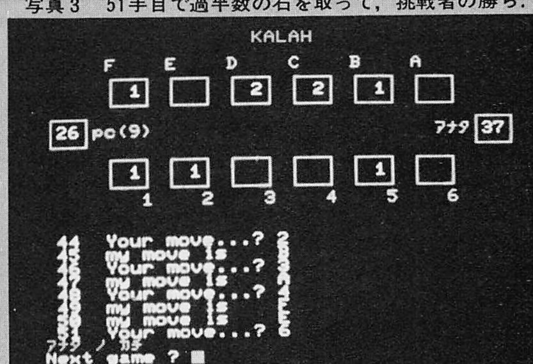


写真4　空のポットを指定すると，注意される．



写真5　マイコン側がじっくり最善手を調べているところ.

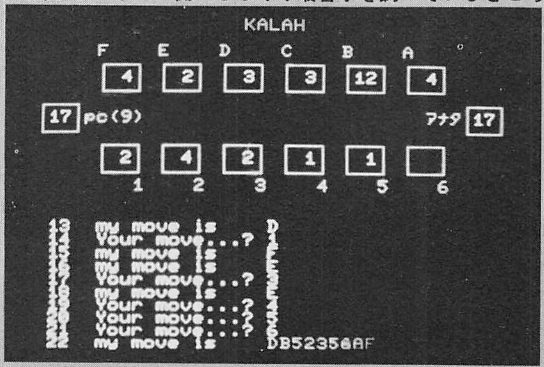


写真6　レベルを6に設定したら，圧勝してしまった….



対戦者はただ見ているだけで良いのです．この間画面の一部がチラつくと思いますが，これは計算機が何を考えているかを表示しているわけなので別に虫ではありません(**写真3**).

人間（つまり対戦者であるあなた）の手番のときは,『あなたの手番』といってくるので，石をばらまきたいポットを決め，その番号1～6を入力します(**写真4**). 石をばらまいたりカラーに移したりする操作はすべて計算機が行なってくれます．もちろん，インチキはしません．

それではこの計算機との知恵比べに果敢にアタックしてください．初めての対戦で計算機に勝てたらあなたは天才です（いうまでもなく筆者は天才ではありませんでした).

3 機種と移植法

書き遅れましたが，このプログラムはPC-8001＋カラーモニタの上で動きます．グリーン・モニタでも問題なく動くと思いますが，その場合は色の違いが濃淡の違いに変わります．また，8001のRAMはフル実装でなくてもOKです．

プログラムの移植法については，これといった簡単な方法はないようで，プログラム・リスト(**リスト1**)を見ながら1行ずつキーインするしかありません．960行以後のDATA文の入力などは非人間的と言っても過言ではないでしょう(プログラム作成者はそれを行なっているわけです). しかし，それでは余りにもサービスが悪すぎるので，工学社でカセット・サービスをしてくれるそうです．詳しくは文末のカセット・サービス欄をご覧ください．

蛇足の極みですが，カセットからプログラムを移植する場合はPC-8001＋モニタTVの他に，カセット再生装置が必要かと思います．PC-8001はクリチカルですから，再生装置には相性の良い機種を選び，再生レベルも最適な点を捜してください．

表1　BASICプログラムの概略

90～ 510	メイン・ルーチン
520～ 640	計算機の手
650～ 690	人間の手
700～ 850	石を動かす
860～ 890	中央部の表示
900～ 950	カラーorポットの表示
960～1500	マシン語

表2　使用される変数とその意味

LV	レベル
S%(0)	人間のカラー
S%(8)～S%(13)	人間のポット
S%(7)	計算機のカラー
S%(1)～S%(6)	計算機のポット
PL	手番，（1＝人間，2＝計算機）
X%(0)～X%(13) Y%(0)～Y%(13)	各ポット，カラーの 画面上の位置
P, P0	ポットの番号(サブルーチンに対するパラメータとしても使用)
T9	ディレイ（値を変えたい方はライン80を変更してください）

表3　マシン語サブルーチンのエントリーアドレス

0035	1文字出力（ROM内）
E000	eval
E030	move（先読みで石を動かす）
E0A0	1文字抹消
E0B0	minimizer
E1C0	maximizer
E2D0	BASICとのインターフェイス(usr0)
E340～E9FF	ワーキング・エリア

4 プログラムの説明

プログラムの主な部分はPC-8001のN-BASICで書かれています．この部分の概略と変数の意味を**表1**，**表2**にそれぞれ示しました．計算機が先読みをして手を決める部分は，高速性を要求されるのでマシン語で書かれています．

マシン語といっても実は**DATA**文としてBASICの中に埋め込まれていますから，その意味では完全なBASICプログラムです．マシン語の部分の主なサブルーチンのエントリーアドレスを**表3**に示します．プログラム中，BASICで書かれた部分で何が行なわれているかは，**リスト1**を読まれればわかると思います．

一方，マシン語で書かれた部分はある意味ではプログラム中最も肝心なところなのですが，この部分について詳しく書くと長くなるので，ここでは計算機が行なっている先読みの戦略の要点のみを述べることにします．

各局面が計算機側から見てどれだけ有利かを数値的に表現するため，局面の評価値を計算機のカラーに入っている石の数から人間のカラーの石の数を引いたものとして定義

図6　2手先までの手番と評価値

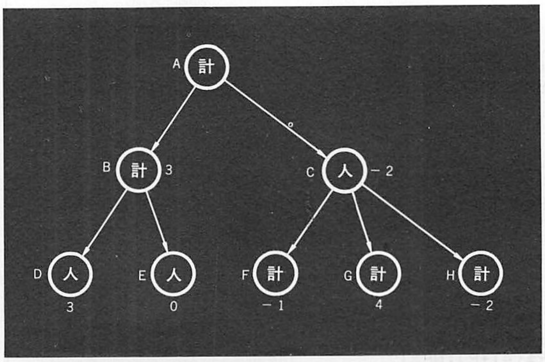
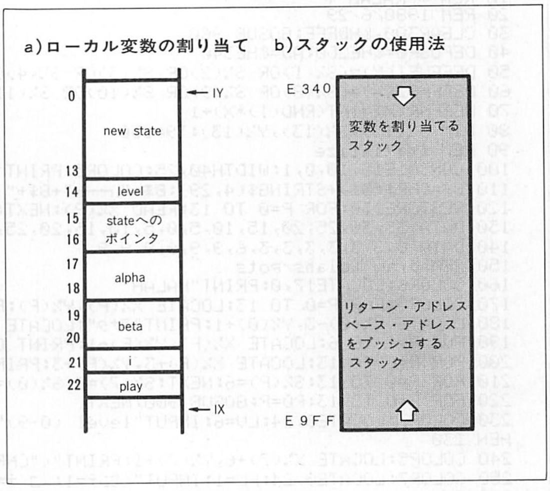


図7　ローカル変数とスタックの関係



します．一度カラーに入った石は減ることがなく，しかも最終的にカラーの石の数によって勝敗が決るのですから，評価値の上のような定義が充分妥当なものであることに異論はないでしょう．

この評価値を計算しているのがevalというサブルーチンで，実際の計算ではポットの石の数も多少考慮しています．計算機はこの評価値が最大になるよう手を選択するべきなので，そのような手を決定するためmaximizerというサブルーチンが用意されています．

他方，人間は評価値が最小になるよう努力するはずですから，その行動を予想するためのminimizerというサブルーチンもあります．これらのサブルーチンが再帰的に他を呼び合いながら，レベルによって指定された回数だけ先読みを行なうのです．

このあたりの動作を具体例によって考えてみます．今，計算機の手番であるとして，そこからすべての可能な手を2手の深さで探索した結果が図6のようだったとします．図中，丸の中に書いてある文字は，どちらの手番かを表わしています，**ルール1**があるため，手番は交互に変わるとは限りません．

さて，計算機の手番A（トップ・レベル．この部分の処理はBASICで書かれています）で，B，Cの2通りの手が選択できます．Bを選ぶと計算機が続けて手を打てます．計算機は評価値が最大になるよう手を選ぶはずなので，ここではmaximizerが呼ばれます．

すると可能な手D，EのうちDが選ばれ，Bの局面における評価の逆算値は結局3になります．次にCを選ぶと手番は人間に移りますが，人間は評価値が最小になるように手を選ぶと考えられるので，ここではminimizerが呼ばれます．

minimizerは可能な手F，G，HのうちHを選び，局面Cの逆算された評価値は−2となります．したがって，計算機はこの2つのうち評価値の大きい方Bを選ぶのです．この方法は**minimax戦略**と呼ばれ，非常に手堅い手作りが得られます．

ところで，minimizerが局面Cを解析する場合，F，G，Hの順に探索を行なうとしますと，実はFの評価値−1が得られた時点で，Cの逆算値が−1以下だということになるので，手Bの方が必ず良いということがわかってしまいます．

ですから，G，Hは探索する必要がなくなります．同様のことはmaximizerの中でも可能で，この考え方による枝刈りは**alpha-beta pruning**と呼ばれています．

この枝刈りによってかなりの探索を省くことができますが，なるべく良い手から探索を行なった方がより多くの枝を刈り取れるので，初めに浅い探索を行なって良さそうな手の見当をつけてからその順に深く探索するという技法がしばしば使われます．

もっとも，カラーのように単純で手の選択の少ないゲームでは，そこまでしなくとも充分高速になるので，今回のプログラムでは単にカラーから遠いポットから探索することにしています．

何手先まで読むかはレベルにより決定されますが，default時のレベル6では最低3手は読みます．その他，自分の手番が続く場合は，そうでない場合より深く読むような細工が施してあります．

最後に，どうしてもマシン語の部分の解読をなさりたい方のために，minimizerおよびmaximizerが使用するローカル変数のスタックへの割り当て法を図7にあげました．

5 あとがき



本文担当のKがこの記事を執筆するにあたり，TK大学S学部IS学科の皆様より数々の助言をいただきました．

特に，Cのオリジナル・プログラムを作成したA．I．氏，それをPC-8001に移植したN．A．氏，また，minimax戦略の解説に関して重要なコメントをくれたK．K．氏の3氏のご援助は本質的です．誌面をお借りしてこれらの方々に厚く御礼申し上げます（都合により氏名はイニシャルのみにさせていただきます）．

なお，本記事に関して，ご意見，ご感想をお持ちの方は，I/O編集部までどしどしお寄せください．

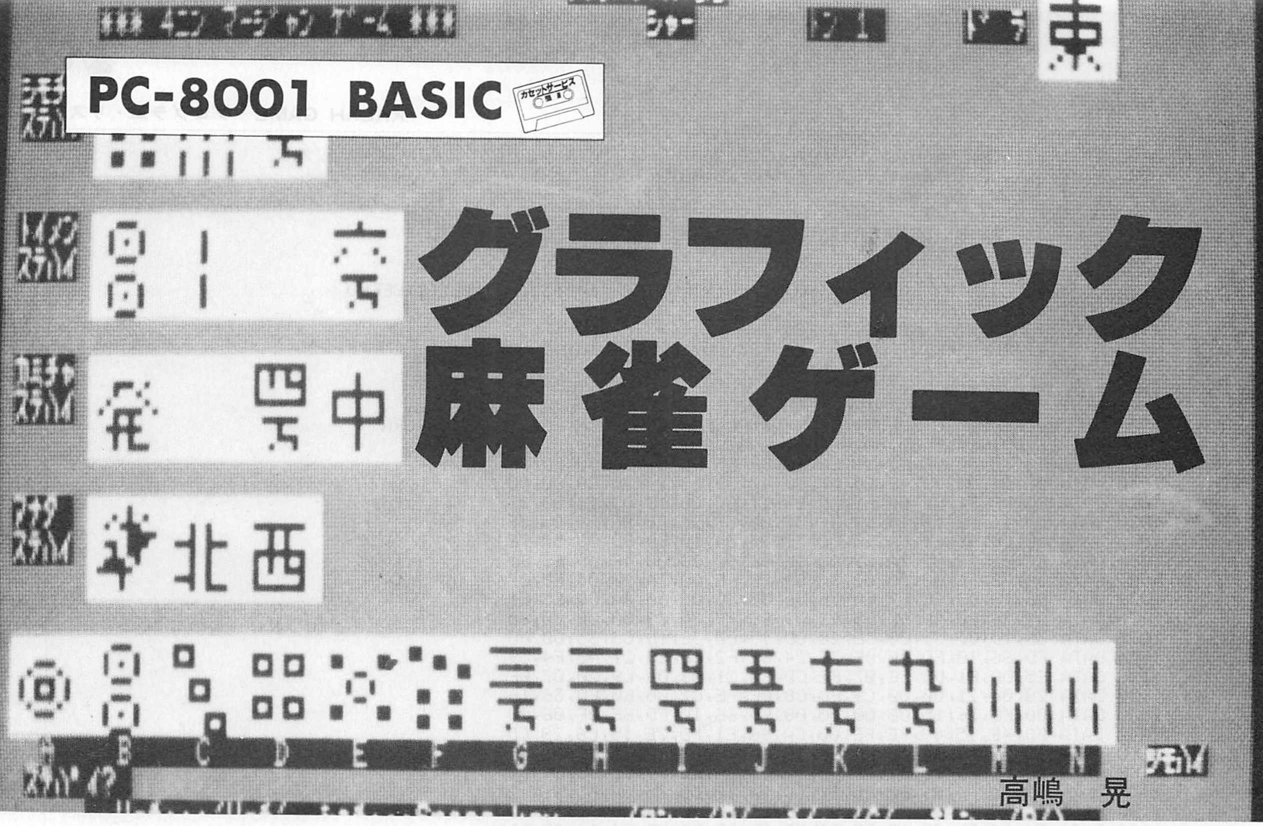




KALAH GAME プログラム・リスト

```
10 REM * KALAH *
20 REM 1980/6/29
30 CLEAR300,&HDFFF:GOSUB 960
40 DEFUSR0=&HE2D0:AD=&HE340
50 DEFFNE1(X)=(S%(1)OR S%(2)OR S%(3)OR S%(4)OR S%(5)OR S%(6))=0
60 DEFFNE2(X)=(S%(8)OR S%(9)OR S%(10)OR S%(11)OR S%(12)OR S%(13))=0
70 DEFFNR(X)=INT(RND(1)*X)+1
80 DIM S%(13),X%(13),Y%(13):T9=150
90 REM initialize
100 CONSOLE15,10,0,1:WIDTH40,25:COLOR5:PRINT CHR$(12)
110 B$=CHR$(31)+STRING$(4,29):B$="┌──┐"+B$+"|  |"+B$+"└──┘"
120 RESTORE130:FOR P=0 TO 13:READ X%(P):NEXT:FOR P=0 TO 13:READ Y%(P):NEXT
130 DATA 35,30,25,20,15,10,5,0,5,10,15,20,25,30
140 DATA 6,3,3,3,3,3,3,6,9,9,9,9,9,9
150 REM draw kalahs/pots
160 COLOR6:LOCATE17,0:PRINT"KALAH"
170 COLOR5:FOR P=0 TO 13:LOCATE X%(P),Y%(P):PRINT B$:NEXT
180 LOCATE X%(0)-3,Y%(0)+1:PRINT"ｱﾅﾀ":LOCATE X%(7)+4,Y%(7)+1:PRINT"pc"
190 FOR P=1 TO 6:LOCATE X%(P),Y%(P)-1:PRINT CHR$(64+P):NEXT
200 FOR P=8 TO 13:LOCATE X%(P)+3,Y%(P)+3:PRINT CHR$(41+P):NEXT
210 FOR P=0 TO 13:S%(P)=6:NEXT:S%(7)=0:S%(0)=0
220 FOR P=0 TO 13:P0=P:GOSUB 900:NEXT
230 COLOR7:LOCATE0,24:LV=6:INPUT"level (0-9)";LV:IF LV<>INT(LV)OR LV<0 OR LV>9 T
HEN 230
240 COLOR5:LOCATE X%(7)+6,Y%(7)+1:PRINT"("CHR$(48+LV)")"
250 COLOR7:LOCATE0,24:PL=1:INPUT"(ｾﾝﾃ=1; ｺﾞﾃ=2)";PL:IF PL<>1 AND PL<>2 THEN 250
260 PN=0
270 REM main loop
280 IF S%(0)>36 OR S%(7)>36 OR(S%(0)=36 AND S%(7)=36)THEN 460
290 IF FNE1(0) OR FNE2(0)THEN 380
300 PN=PN+1:COLOR7:LOCATE0,24:PRINT PN;
310 ON PL GOTO 350,320
320 GOSUB 520:GOSUB 700 'computer
330 IF P=7 THEN PL=2 ELSE PL=1
340 GOTO 270
350 GOSUB 650:GOSUB 700 'man
360 IF P=0 THEN PL=1 ELSE PL=2
370 GOTO 270
380 REM pot empty
390 BEEP1:FOR T=1 TO T9:NEXT:BEEP0
400 COLOR6:LOCATE0,24:O$="ﾉ pot ﾊ ｾﾞﾝﾌﾞ ｶﾗｯﾎﾟ"
410 IF FNE1(0) THEN PRINT"ﾜﾀｼ"O$
420 IF FNE2(0) THEN PRINT"ｱﾅﾀ"O$
430 FOR T=1 TO T9*10:NEXT
440 FOR P=1 TO 6:K0=7:P0=P:GOSUB 820:NEXT P
450 FOR P=8 TO 13:K0=0:P0=P:GOSUB 820:NEXT P
460 REM end of game
470 LOCATE0,24:COLOR6
480 IF S%(0)>36 THEN PRINT "ｱﾅﾀ ﾉ ｶﾁ"
490 IF S%(7)>36 THEN PRINT "ﾜﾀｼ ﾉ ｶﾁ"
500 IF S%(0)=S%(7)THEN PRINT "ﾋｷﾜｹ"
510 BEEP:COLOR7:INPUT"Next game ";O$:GOTO 90
520 REM computer's move
530 DIM R%(6):LOCATE5,24:PRINT"my move is      ";
540 M=-3000:P=0:FOR R=1 TO 6:COLOR R
550 P0=FNR(6):IF R%(P0) THEN 550
560 R%(P0)=1:IF S%(P0)=0 THEN 590
570 FOR I=0 TO 13:POKE AD+I,S%(I):NEXT:POKE AD+14,LV:POKE AD+15,P0
580 PRINT CHR$(64+P0);:V=USR0(CINT(M)):PRINT CHR$(29);:IF V>M THEN M=V:P=P0
590 NEXT R:ERASE R%:IF P THEN 610
600 P=FNR(6):IF S%(P)=0 THEN 600
610 COLOR7:PRINT CHR$(64+P)
620 BEEP1:FOR T=1 TO 10:NEXT:BEEP0
630 COLOR2:LOCATE X%(P)+1,Y%(P)+1:PRINT USING"##";S%(P);
640 FOR T=1 TO T9*10:NEXT:RETURN
650 REM man's move
660 P=0:COLOR 7:LOCATE5,24:INPUT"Your move...";P
670 IF P<>INT(P) OR P<1 OR P>6 THEN COLOR2:PRINT TAB(5)"1 ｶﾗ 6 ﾏﾃﾞ ﾉ ｾｲｽｳ ｦ ｲﾚﾙ
ﾍﾞｼ":GOTO 660
680 IF S%(P+7)=0 THEN COLOR2:PRINT TAB(5)"ｿﾉ pot ﾊ ｶﾗｯﾎﾟ !":GOTO 660
690 P=P+7:RETURN
700 REM move
710 COLOR2:LOCATE X%(P)+1,Y%(P)+1:PRINT USING "##";S%(P);
720 P0=P:GOSUB 860
730 K1=-(P>7)*7:K0=7-K1
740 S=S%(P):GOSUB 860:P0=P:S%(P0)=0:GOSUB 900
750 P=(P+1)MOD 14:IF P=K1 THEN P=P+1
760 S=S-1:GOSUB 860:P0=P:S%(P0)=S%(P0)+1:GOSUB 900
770 IF S THEN 750
780 IF P=K0 THEN RETURN
790 IF S%(P)>1 OR S%(14-P)=0 THEN RETURN
800 IF(K0=0 AND P<7)OR(K0=7 AND P>7)THEN RETURN
810 P0=P:GOSUB 820:P0=14-P
820 REM add to kalah
830 IF S%(P0)=0 THEN RETURN
```

```
840 S=S%(P0):GOSUB 860:S%(P0)=0:GOSUB 900
850 S%(K0)=S%(K0)+S:S=0:GOSUB 860:P0=K0:GOTO 900
860 REM display (center)
870 COLOR7:LOCATE18,7
880 IF S THEN PRINT USING "##";S ELSE PRINT "  "
890 RETURN
900 REM display (pot/kalah)
910 COLOR2:LOCATE X%(P0)+1,Y%(P0)+1:PRINT USING "##";S%(P0):BEEP1:BEEP0
920 FOR T=1 TO T9:NEXT
930 COLOR7:LOCATE X%(P0)+1,Y%(P0)+1
940 IF S%(P0)THEN PRINT USING "##";S%(P0) ELSE PRINT "  "
950 RETURN
960 REM poke subroutines
970 PRINT "Wait a moment":RESTORE 990
980 FOR AD=&HE000 TO &HE33F:READ O$:POKE AD,VAL("&H"+O$):NEXT:RETURN
990 DATA E5,DD,E1,3E,00,06,06,DD,86,01,DD,96,08,DD,23,10
1000 DATA F6,16,00,5F,F2,19,E0,16,FF,E5,DD,E1,26,00,DD,6E
1010 DATA 07,06,00,DD,4E,00,A7,ED,42,29,29,29,19,C9,00,00
1020 DATA E5,DD,E1,FE,07,F2,3D,E0,11,07,00,18,03,11,00,07
1030 DATA D5,16,00,5F,19,D1,46,36,00,3C,23,FE,0E,20,05,3E
1040 DATA 00,DD,E5,E1,BA,20,02,3C,23,34,10,ED,BB,C8,57,7E
1050 DATA FE,01,28,02,7A,C9,00,3E,0E,92,06,00,4F,DD,E5,E1
1060 DATA 09,7E,FE,00,20,02,7A,C9,7B,FE,00,7A,20,05,FE,07
1070 DATA F8,18,03,FE,07,F0,7E,36,00,3C,06,00,4A,DD,E5,E1
1080 DATA 09,36,00,4B,DD,E5,E1,09,86,77,7A,C9,00,00,00,00
1090 DATA 3E,1D,CD,35,00,3E,20,CD,35,00,3E,1D,C3,35,00,00
1100 DATA FD,66,10,FD,6E,0F,3E,24,BE,F2,C0,E0,21,48,F4,C9
1110 DATA E5,DD,E1,DD,BE,07,F2,CD,E0,21,B8,0B,C9,20,07,BE
1120 DATA 20,04,21,00,00,C9,FD,CB,0E,7E,C2,00,E0,FD,36,16
1130 DATA 00,FD,36,15,08,00,00,00,FD,66,10,FD,6E,0F,06,00
1140 DATA FD,4E,15,09,7E,FE,00,CA,96,E1,FD,7E,15,C6,29,CD
1150 DATA 35,00,FD,34,16,FD,66,10,FD,6E,0F,FD,E5,D1,01,0E
1160 DATA 00,D5,ED,B0,E1,FD,7E,15,CD,30,E0,FD,E5,DD,E1,01
1170 DATA 17,00,DD,09,FD,E5,E1,DD,74,10,DD,75,0F,FD,66,12
1180 DATA FD,6E,11,DD,74,12,DD,75,11,FD,66,14,FD,6E,13,DD
1190 DATA 74,14,DD,75,13,FE,00,FD,7E,0E,FD,E5,20,10,3D,DD
1200 DATA 77,0E,DD,E5,FD,E1,CD,B0,E0,18,0F,00,00,00,D6,02
1210 DATA DD,77,0E,DD,E5,FD,E1,CD,C0,E1,FD,E1,CD,A0,E0,54
1220 DATA 5D,FD,46,14,FD,4E,13,A7,ED,42,F2,96,E1,62,6B,FD
1230 DATA 46,12,FD,4E,11,A7,ED,42,F2,8D,E1,EB,C9,FD,72,14
1240 DATA FD,73,13,00,00,00,FD,34,15,FD,7E,15,FE,0E,FA,E8
1250 DATA E0,FD,35,16,FA,AE,E1,FD,66,14,FD,6E,13,C9,FD,7E
1260 DATA 00,FE,24,F2,BA,E1,21,B8,0B,C9,21,00,00,C9,00,00
1270 DATA FD,66,10,FD,6E,0F,3E,24,BE,F2,D0,E1,21,48,F4,C9
1280 DATA E5,DD,E1,DD,BE,07,F2,DD,E1,21,B8,0B,C9,20,07,BE
1290 DATA 20,04,21,00,00,C9,FD,CB,0E,7E,C2,00,E0,FD,36,16
1300 DATA 00,FD,36,15,01,00,00,00,FD,66,10,FD,6E,0F,06,00
1310 DATA FD,4E,15,09,7E,FE,00,CA,A6,E2,FD,7E,15,C6,40,CD
1320 DATA 35,00,FD,34,16,FD,66,10,FD,6E,0F,FD,E5,D1,01,0E
1330 DATA 00,D5,ED,B0,E1,FD,7E,15,CD,30,E0,FD,E5,DD,E1,01
1340 DATA 17,00,DD,09,FD,E5,E1,DD,74,10,DD,75,0F,FD,66,12
1350 DATA FD,6E,11,DD,74,12,DD,75,11,FD,66,14,FD,6E,13,DD
1360 DATA 74,14,DD,75,13,FE,07,FD,7E,0E,FD,E5,20,10,3D,DD
1370 DATA 77,0E,DD,E5,FD,E1,CD,C0,E1,18,0F,00,00,00,D6,02
1380 DATA DD,77,0E,DD,E5,FD,E1,CD,B0,E0,FD,E1,CD,A0,E0,54
1390 DATA 5D,FD,66,12,FD,6E,11,A7,ED,52,F2,A6,E2,00,00,FD
1400 DATA 66,14,FD,6E,13,A7,ED,52,F2,9D,E2,EB,C9,FD,72,12
1410 DATA FD,73,11,00,00,00,FD,34,15,FD,7E,15,FE,07,FA,F8
1420 DATA E1,FD,35,16,FA,BE,E2,FD,66,12,FD,6E,11,C9,FD,7E
1430 DATA 07,FE,24,F2,CA,E2,21,48,F4,C9,21,00,00,C9,00,00
1440 DATA ED,73,FE,E9,31,FE,E9,FD,21,40,E3,FD,E5,E1,FD,7E
1450 DATA 0F,CD,30,E0,FD,E5,DD,E1,01,10,00,DD,09,FD,E5,E1
1460 DATA DD,74,10,DD,75,0F,2A,A8,F0,00,00,00,DD,74,12,DD
1470 DATA 75,11,DD,36,14,0B,DD,36,13,B8,FE,07,FD,7E,0E,FD
1480 DATA E5,20,0C,DD,77,0E,DD,E5,FD,E1,CD,C0,E1,18,0C,D6
1490 DATA 02,DD,77,0E,DD,E5,FD,E1,CD,B0,E0,FD,E1,22,A8,F0
1500 DATA E1,F9,C9,00,00,00,00,00,00,00,00,00,00,00,00,00
```



PC-8001 BASIC

グラフィック麻雀ゲーム

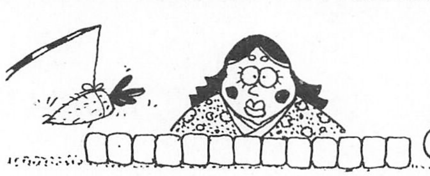
高嶋　晃

おっ，おっ，面白い！　APPLEの麻雀ゲームはすばらしい！　しかし，私にはAPPLEを購入できるほどの財産もない．打ちひしがれていたときに突如彗星の如く現われたPC-8001．カラーグラフィックと24K BASICをひっさげて，しかも超安価．『これだ！』と思って，即購入したのですが……，ＰＣの特長を生かした麻雀ゲームがない．

そこで，待ち切れず奮起して作ったのが，この『グラフィック麻雀ゲーム』です．どうやら苦労のかいあってAPPLE並みの（『もしやAPPLE以上では……』と自負しているのですが）ものができたので発表します．

このゲームは160×100ドット・グラフィックをフルに生かしたため，家庭用テレビを使用の方は少々字が読みづらいかと思いますが，そこは想像力で補い，楽しんでください．

なお，このプログラムは32K RAMをほとんど使用してしまいます．したがって，標準ＲＡＭ実装のみの方は残念ながら写真を眺めているか，ＲＡＭを追加するかのどちらかです．しかし，このゲームで練習すれば，ＲＡＭ代ぐらい後日麻雀で稼げるのでは……．



ゲームの特長

❶牌は10×12ドットのグラフィックで表示されます．

❷カラーをふんだんに使い，配牌時もあきないようにデモンストレーションを表示します．

❸ロンすると自動的に役判定および点数計算を行なってくれます．

❹持ち点は27,000点，東々回しの半荘シメで30,000点返しです．半荘が終了すると最終合計が表示されてＥＮＤとなります．

❺連荘ありで，リーチするとリーチ代として1,000点減ります．なお，連荘時の加算点と二飜しばりは行なっていません．

❻４人麻雀であることやポン，チーができることなど，その他はI/O'80年６月号のＰＣ用４人麻雀ゲームと同様です．しかし，デバッグや改良を行なっています．



遊び方

ＲＵＮすると『フルグラフィック　マージャンゲーム』のタイトルが現われます(写真１)．約３秒程度で画面が一転して，簡単なゲームの説明を表示し，他家の強さを聞いてきます(写真２)．[1]，[2]，[3]のいずれかを入力してください．

ＰＣが初期条件を決め，配牌，洗牌を行なっている間は牌が次々と表示されます(写真３)．なお，この牌は配牌とは何の関係もありません．それが終わるとサイが乱数で振られ，あなたの場所が決定されます(写真４)．

次に，メインの画面が表示され，東家から順々にツーモ，捨て牌を行なっていきます．そして，あなたのツーモが済むと捨て牌を聞いてくるので，不要牌の下のコードＡ～Ｎのいずれかを入力します(写真５)．入力した牌は河(ホー)に捨て

写真1　ゲーム・タイトルがしばらく表示される．



写真2　ゲームの説明．ここで強さを入力する．

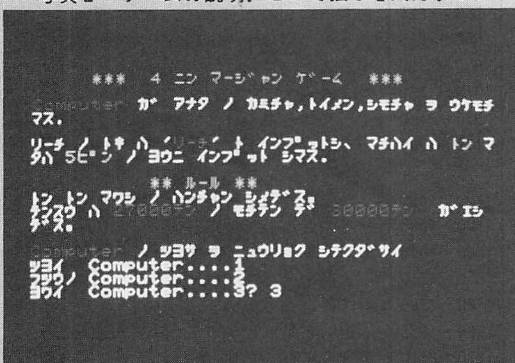


写真3　配牌，洗牌のときは次々と牌が表示される．

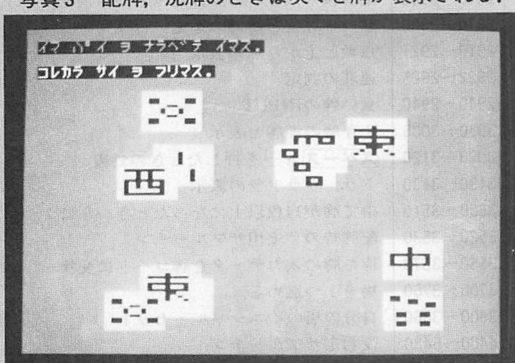


写真4　場所の決定．

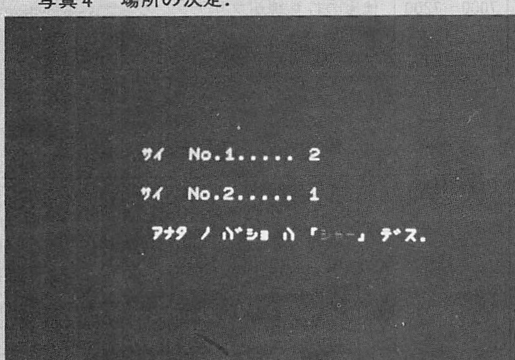


写真5　不要牌はA～Nで指定

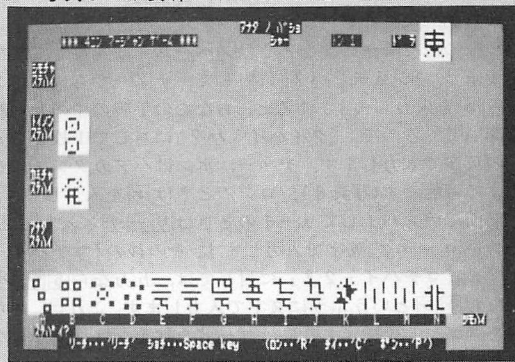


写真6　リーチをかけて，マチハイを入力しているところ．

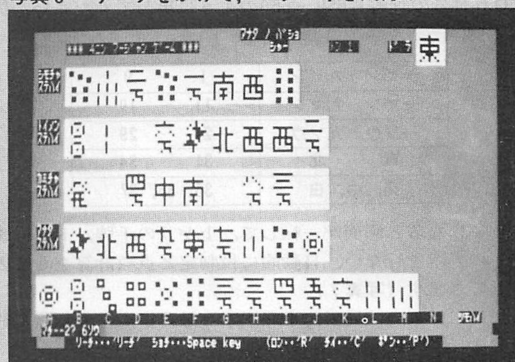


写真7　ロンになると役，点数計算も行なわれる．

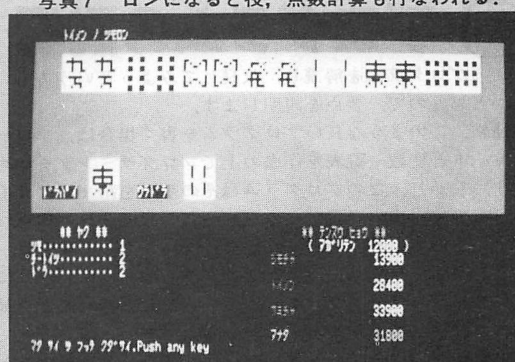
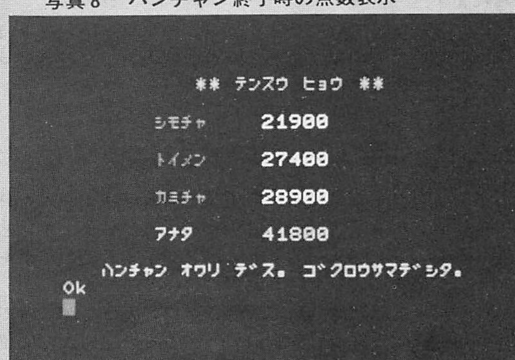


写真8　ハンチャン終了時の点数表示



られ，その後，他3家のツーモ牌，捨て牌，ロンのチェックが自動的に行なわれます．

他家の捨て牌が必要なときは[スペース]キーを押して処置を要求し，『**ショチ　ハ？**』に対して，チィーなら[c]，ポンなら[p]を入力します．すると，あなたの手牌の中の食い牌を聞いてくるので，『**クイハイ　ハ？**』に対して1枚ずつA～の英字で入力します．チィー，ポンはハダカタンキになるまで可能です(**写真6**)．ロンのときは[r]を入力します．

面前でテンパイしてリーチのときは[リ][ー][チ]と入力し，捨て牌を[a]～[n]の英字で入力します．その後の『**マチハイ－1？**』，『**マチハイ－2？**』に対しては，五万なら[5][マ][ン]，東なら[ト][ン]というふうに文字で入力します．これは，待ち牌を牌コードで指定するにはコードを覚えなければならないし，味気ないと思ったからです．

カナ文字の入力は面倒だという方は，行番号650～662をI/O 6月号の行番号650～660のように戻してください．従来の牌コードで入力できます(**表1**)．

一ピン～九ピン	1　～　9
一マン～九マン	11　～　19
一ソウ～九ソウ	21　～　29
東　～　北	31　～　34
発，中，白	35，36，37

ロンになると画面が一転して，上がりの手牌，ドラ，裏ドラの表示を行ない，自動的に役判定と点数計算をして表示してきます(**写真7**)．

場は常に東の○局で，半荘すると最終得点を表示して終了です(**写真8**)．

プログラムの説明

このプログラムを解読し，改良してみようという方もいるかと思うので，要点を説明します．

通常，このような長いプログラムを書く場合は，フローチャートと変数一覧表を作成の上でプログラミングを始めます．しかし，このプログラムはその手順を踏まずに，I/O 6月号の峰岸氏の麻雀ゲームをベースとしてそれに肉をつけていく形で追加，改良，修正を行なったものです．

したがって，でき上がったプログラムは非常にわかりにくいスパゲッティ・プログラムとなってしまったことを最初にお詫びしておきます．

プログラムは，ゲーム進行を行なうメイン・ルーチンと，ロン後の役判定，点数計算サブルーチンの2つに大きく分けられます．

図1，2，3のフローチャート（このフローチャートは完成後に書いたものです）と，変数一覧表および行番号マップを**表2**，**表3**に示します．

メイン・ルーチン

基本的なロジックは峰岸氏のTK用4人麻雀ゲームをPC用に移植したもの（以下，『移植ゲーム』と書きます）と同様なので，I/Oの'79年12月号および'80年6月号の記事を参照してください．以下は改良点について述べます．

37行までは一ピンから白までの各々の牌データが入っており，40行，80行で必要な牌のデータのみを配列A%に読み出し，85行の**PUT**@文で表示しています．このため，グラフィック牌の配列は1個分のA%（8）で済みました．

表2　行番号マップ

行番号	内　　容
1～ 37	牌のグラフィック・データ．牌コードと行番号が対応
40～ 85	牌の表示サブルーチン．A（8）に牌データが入る．
100～ 107	初期設定
110～ 115	乱数のRANDMIZE化
120～ 145	ゲームの説明表示および強さのINPUT
270～ 305	自分のリーチのロン・チェック
310～ 320	マイコンの強さ決定
350～ 370	リーチ宣言の決定．P（J1）とQ（J1）によって決まる．
400～ 460	ロンとナガレのチェック．
470～ 550	捨て牌の表示と2秒間の持ち時間カウント
560～ 575	リーチ時の捨て牌の表示
580～ 625	他者のロンのチェック
650～ 665	リーチの待ち牌のINPUTとリーチの表示
670～ 750	埋牌
810～ 950	ロンの表示
970～1150	上がり牌の表示
1170～1280	下家，対面，上家の各々のロン・チェック．Q1＝1ならロン
1290～1490	ツーモ牌の表示および捨て牌の処置
1500～1595	サイの目と場所の表示
1600～1640	ドラ，4人の河，自分の牌などの表示
1650～1710	順子選定
1720～1750	対子選定
1820～1865	変数，配列のクリア
1870～2410	下家，対面，上家の配牌
2420～2460	自分の配牌
2470～2560	配牌後の残り牌の読み込みとかき混ぜ．
2570～2600	リーチ宣言と時期の選定
2910～2921	点数と上がり点の表示
2922～2925	連荘の判定
2940～2940	食い牌のINPUT
3030～3085	ナキ牌の埋牌と表示
3090～3120	[スペース]キーを押したときの処置
3130～3130	ドラ，ウラドラの表示
3500～3510	捨て牌が14枚以上になったとき，左に1つ移動
3520～3530	配牌時のデモ用サブルーチン
3550～3670	待ち牌の入力データを牌コードに変換
3700～3760	場を1つ進める．
3800～3860	自分の場のイニシャル・セット
4400～5480	役判定サブルーチン
5500～6120	点数計算サブルーチン
6200～6310	半荘終了後の最終点数計算
7000～7200	他家の待ち牌がフリテンになっているか否かの判定
8500～8770	タイトル表示サブルーチン

表3　主要変数一覧表

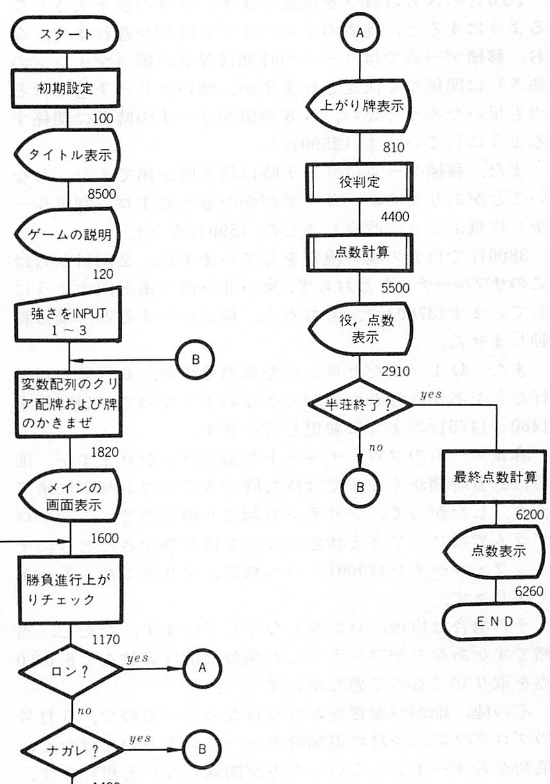
C(J1, I1)	4家の配牌 J1＝1は下家，2，3，4は対面，上家，自分です．I1＝1～13は配牌，14はツーモ牌，15～16は待ち牌です．
P1	ツモロンした家
P2	振り込みのとき，和了(ホーラ)した家
P3	振り込みのとき，振り込んだ家
P5	親の家
P6	自分のリーチのとき，P6＝4

P 8(I)	リーチ後の各々のツーモ回数
L, E(84)	ツーモ回数およびツーモ牌
N 4	自分のナキ回数
G (I)	ナキ牌
RE	連チャン回数
H 5	上がった家の風．1＝東．2，3，4は各々南，西，北です．
F (4)	各家の点数
FR	場のリーチ代

以上のP 1～P 6およびP 8，G，Fの各配列添字Iは1＝下家，2＝対面，3＝上家，4＝自分です．続いて，以下に役判定および点数計算サブルーチンで使用している変数を列記します．

K (I)	手牌の情報
K (1)	最終ツーモ牌
K (4)～K (7)	ピンズ，マンズ，ソウズ，字牌の各々枚数
K (8)	頭のある位置で，4～7の数字が入っており，K (4)～K (7)を示します．
K (10)～K (13)	順子の先頭の位置の牌コードが入っています．
K (14)～K (17)	刻子の牌コードが入っています．
K (18)	頭の牌コードが入っています．
L (1)～L (37)	手牌の牌コード別の枚数
E (1)～E (14)	上がった家の手牌．C (J 1，I 1)をソーティングして移し替え
M	役満の数
N	飜数
R－10	順子の数
S－14	刻子の数
R 1	チィーをした回数
S 1	ポンをした回数
Z 1	基礎点
Z 3	点数

図1　概略フローチャート



※右下の数字は行番号を示します．

図2　役判定フローチャート

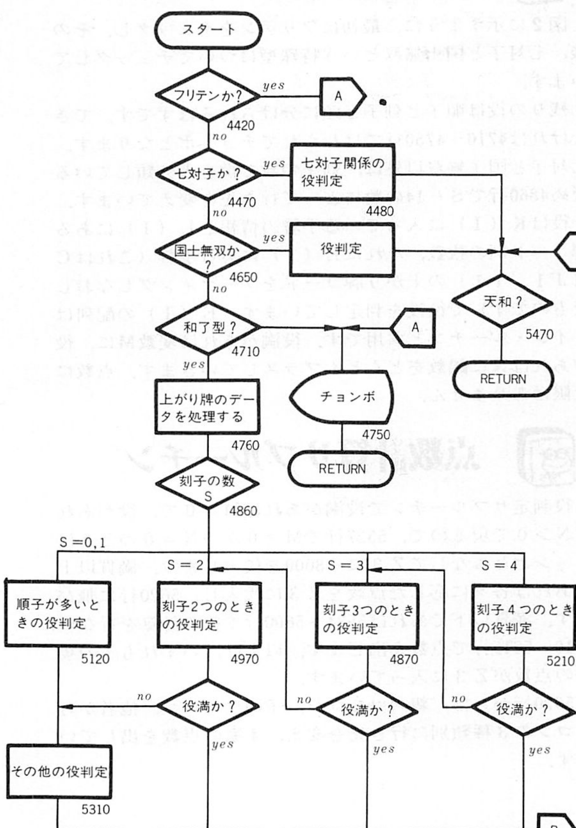
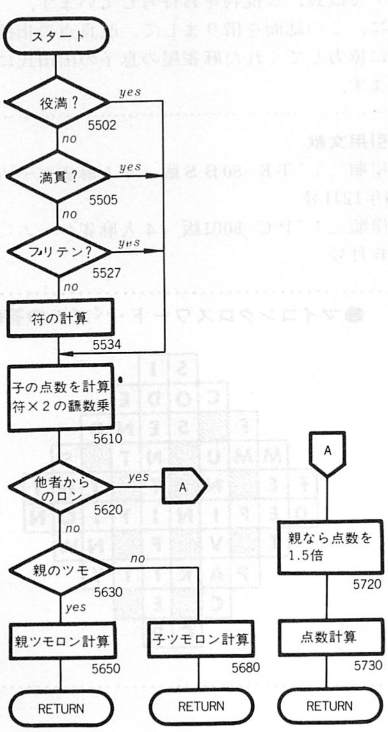


図3　点数計算フローチャート



150行のＮ８は強さを決定します．Ｎ８の値を大きくなるようにすると，他者のテンパイする時期が遅れます．なお，移植ゲームではリーチの時期はＮ８の値（マイコンの強さ）に関係なく決定されますが，強いとリーチをかけるのも早いだろうと思い，Ｎ８の値がリーチの時期に関係するようにしています（2590行）．

また，移植ゲームはリーチ時に待ち牌が出てもロンしないことがありますが，リーチがかかると必ずロン判定ルーチンに飛ぶように改良しました（590行など）．

3800行で自分の場の決定をしていますが，２回目からはこのサブルーチンをとおらず，東→北→西→南と回すようにしています(3700行)．もちろん，親がロンすると，場は移動しません．

また，Ｑ１＝１だとロンになるのですが，６月号の1460行だと上者からしかロンができないようなので，1460行を1460～1475行のように変更しています．

麻雀ゲームのフローチャートを追えばわかりますが，他者はある時期がくるまでは待ち牌がきてもロンせずに捨てます．したがって，フリテンも起こり得るので，このプログラムではロンできる状態になって待ち牌がきたときにチェック・ルーチン(7000行～)へ飛び，フリテンならＸ＝１で戻ります．

その場合は以後，ロンをしなくしています．ただし，当然ですがあなたがフリテンした場合は情け容赦なく8,000点を取り立てるので悪しからず……．

その他，細かい変更をかなり行なっているので，６月号のプログラムに今月の追加分をキーインするのではなく，最初からキーインしていった方が間違いないと思います．

役判定サブルーチン

図２に示すように，最初にフリテンをチェックし，その後，七対子と国士無双という特殊型についてチェックしています．

残りの役は順子(シュンツ)と刻子(コーツ)と頭に分けられるはずです．できなければ4710～4750行ではねられてチョンボとなります．七対子と国子無双以外は，刻子の数で役を大分類しているため4860行でＳ－14の数によって行き先を変えています．

役はＫ（Ｉ）に入っている手牌の情報とＬ（Ｉ）にある牌コード別の枚数，それにＥ（Ｉ）にある手牌（これはＣ（Ｊ１，Ｉ１）の上がり牌コードをソーティングしなおしたものです）で役役を判定しています．Ｅ（Ｉ）の配列はメイン・ルーチンと共用です．役満があれば変数Ｍに，役があればＮに飜数をどんどんプラスしていきます．点数に上限はありません．

点数計算サブルーチン

役判定サブルーチンで役満があればＭ＞０で，役があればＮ＞０で戻るので，5527行でＭ＝０かつＮ＝０のときはチョンボとみなしてＺ３に－8000を代入します．満貫以上であれば各々に応じた点数をＺ３に代入し，5620行に飛びます．満貫以下であれば5534～5600行で符の計算を行ない，5610～5615行で点数を出します．以上は，いずれも子の場合の点数がＺ３に入っています．

5620行からは，親のツモロン，子のツモロン，他者からのロンの３種類別に行き先を変え，４家の点数を出しています．

今後の改良点

特長の項で，『連荘時の加算点と二飜しばりはない』と述べましたが，これは簡単にできると思うので，ぜひ挑戦してみてください．連荘回数は変数ＲＥに入っています．

その他に，ノーテンでは絶対にロンさせないようにするとか，他者にもナキを認めるなどの改良をすれば，より実戦らしくなるでしょう．ただし，画面表示位置やロジックの変更を余儀なくされるので，少々手間がかかると思います．

ＰＣのカラーグラフィックについて

ＰＣはドット単位で色が決められないため中，発，一ソウなどの文字に色がつけられませんでした．また，このプログラムではバックカラーを緑のヌルドット（color 4，255，１：chr＄（12））で，牌を10×12ドットのリバース（put@……not）で表示したため，表示位置がずれるとその欠点がでてきます．

つまり，表示位置をずらすと牌が大きくなったり，牌の周辺が黒くなったりします．牌の大きさも表示位置も，必ずキャラクタ単位の枠にはまるようにしなければなりません．

とはいうものの，その制約もＰＣのＲＢＧ入力の色の美しさと24K BASICの多機能さが，これを補って余りあると思います．

あとがき

短かいプログラムは除くとして，長いプログラムになるとバグを完全になくすというのは難かしく，これも考え違いやつけ忘れなどのバグがあるかも知れません．皆さんのデバッグや改良，ご批判をお待ちしています．

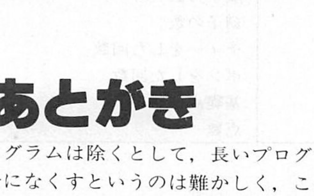
最後に，この誌面を借りまして，改良点の指摘やテスト・ランに協力してくれた麻雀屋の息子の山田氏にお礼を申し上げます．

参考引用文献

1) 峰岸順二：〝ＴＫ-80ＢＳ版　４人麻雀ゲーム〟，I/O，'79年12月号

2) 峰岸順二：〝ＰＣ-8001版　４人麻雀ゲーム〟，I/O，'80年６月号

●マイコンクロスワード・パズルの答●



グラフィック 麻雀ゲーム　プログラム・リスト

```
1 DATA 120,0,0,17422,-23832,11914,-8124,0,0
2 DATA 120,-8192,1088,17429,224,3584,20548,1089,14
3 DATA 120,14336,-32608,3,-32544,3586,-32768,2563,56
4 DATA 120,0,-28960,-4566,0,15232,-18262,3,0
5 DATA 120,0,-26528,12385,-32480,3076,6534,6,0
6 DATA 120,12288,3267,0,-15568,12,12492,3,0
7 DATA 120,6144,864,-32660,-16383,13068,12288,3267,0
8 DATA 120,12288,3267,-13312,816,13056,204,3264,51
9 DATA 120,0,-25760,109,26328,27,-9802,6,0
11 DATA 120,0,-32768,63,0,7936,-16368,1153,0
12 DATA 120,-4096,1,-512,0,7936,-16368,1153,0
13 DATA 120,-2048,3,31,1016,7936,-16368,1153,0
14 DATA 120,-2048,-30045,-32198,1016,7936,-16368,1153,0
15 DATA 120,-4096,257,20510,1016,7936,-16368,1153,0
16 DATA 120,16384,4064,17408,520,7936,-16368,1153,0
17 DATA 120,8192,4064,2050,992,7936,-16368,1153,0
18 DATA 120,0,320,16905,512,7936,-16368,1153,0
19 DATA 120,8192,2016,18450,800,7936,-16368,1153,0
21 DATA 120,-32768,2688,-8163,-31823,5894,16446,256,0
22 DATA 120,0,256,4100,64,1024,16400,256,0
23 DATA 120,0,256,4100,64,4352,4164,1089,0
24 DATA 120,0,1088,17425,272,4352,4164,1089,0
25 DATA 120,0,-30688,-28128,584,9345,2194,2082,0
26 DATA 120,0,1344,21525,336,5376,20564,1345,0
27 DATA 120,16384,256,4,8776,9353,18432,-30430,36
28 DATA 120,0,-26528,12882,1033,16512,10546,6245,0
29 DATA 120,18432,-30430,36,8776,9353,18432,-30430,36
31 DATA 120,0,-32512,4159,9208,16265,20536,2337,0
32 DATA 120,0,-32512,4159,9208,11912,-5998,2338,0
33 DATA 120,0,-32768,10303,-8032,10895,2282,4066,0
34 DATA 120,0,0,10250,-31816,2562,-18392,3712,0
35 DATA 120,0,0,18478,-24048,2583,-24452,3712,0
36 DATA 120,0,256,-508,8776,16265,16400,256,0
37 DATA 120,0,0,0,0,0,0,0,0
38 GOTO100
40 ONC(J1,I1)GOTO41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,
62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77
41 RESTORE1:GOTO80
42 RESTORE2:GOTO80
43 RESTORE3:GOTO80
44 RESTORE4:GOTO80
45 RESTORE5:GOTO80
46 RESTORE6:GOTO80
47 RESTORE7:GOTO80
48 RESTORE8:GOTO80
49 RESTORE9:GOTO80
51 RESTORE11:GOTO80
52 RESTORE12:GOTO80
53 RESTORE13:GOTO80
54 RESTORE14:GOTO80
55 RESTORE15:GOTO80
56 RESTORE16:GOTO80
57 RESTORE17:GOTO80
58 RESTORE18:GOTO80
59 RESTORE19:GOTO80
61 RESTORE21:GOTO80
62 RESTORE22:GOTO80
63 RESTORE23:GOTO80
64 RESTORE24:GOTO80
65 RESTORE25:GOTO80
66 RESTORE26:GOTO80
67 RESTORE27:GOTO80
68 RESTORE28:GOTO80
69 RESTORE29:GOTO80
71 RESTORE31:GOTO80
72 RESTORE32:GOTO80
73 RESTORE33:GOTO80
74 RESTORE34:GOTO80
75 RESTORE35:GOTO80
76 RESTORE36:GOTO80
77 RESTORE37
80 FORO=0TO8:READA%(O):NEXT
85 COLOR7:PUT@(X,Y)-(X+9,Y+11),A%,NOT:RETURN
100 DEFINTB,D-E,G-J,L-Y:DIMA(8),B(8),R$(14),C(4,16),D(37),E(84),G(12),M(4,21),P(
4),P8(4),Q(4),K(18),L(37),F(4):U$="ｼﾓﾁｬ":V$="ﾄｲﾒﾝ":W$="ｶﾐﾁｬ":X$="ｱﾅﾀ ":Y$="ｽﾃﾊｲ"
102 GOSUB8500
104 PRINTCHR$(12):P5=0:RE=0:N5=0
105 CONSOLE,,0,1:WIDTH40,25
107 DEFFNA(Y)=INT(Y/10):DEFFNB(Y)=Y-10*INT(Y/10)
110 LOCATE0,9:PRINT"ﾊﾟｲ ｦ ｶｷﾏｾﾞﾙﾀﾒ ";:COLOR2:PRINT"5-10 秒 ";:COLOR7:PRINT"ﾉﾁﾆ":P
RINT:COLOR6:PRINT"Space key";:COLOR7:PRINT" ｦ ｵｼﾃｸﾀﾞｻｲ."
115 A$=INKEY$:P7=RND(1):IFA$<>" "THEN115
120 PRINTCHR$(12):COLOR4:PRINT"      ***  4 ﾆﾝ ﾏｰｼﾞｬﾝ ｹﾞｰﾑ  ***":PRINT
121 COLOR2:PRINT"Computer";
122 COLOR7:PRINT" ｶﾞ ｱﾅﾀ ﾉ ｶﾐﾁｬ,ﾄｲﾒﾝ,ｼﾓﾁｬ ｦ ｳｹﾓﾁﾏｽ.":PRINT
123 PRINT"ﾘｰﾁ ﾉ ﾄｷ ﾊ ｢";:COLOR2:PRINT"ﾘｰﾁ";:COLOR7:PRINT"｣ ﾄ ｲﾝﾌﾟｯﾄｼ､ ﾏﾁﾊｲ ﾊ ";:
COLOR6:PRINT"ﾄﾝ";:COLOR7:PRINT" ﾏﾀﾊ ";:COLOR6:PRINT"5ﾋﾟﾝ";:COLOR7:PRINT" ﾉ ﾖｳﾆ ｲ
ﾝﾌﾟｯﾄ ｼﾏｽ.":PRINT
125 LOCATE10,9:COLOR4:PRINT"** ﾙｰﾙ **":COLOR7:PRINT"ﾄﾝ ﾄﾝ ﾏﾜｼ ﾉ ﾊﾝﾁｬﾝ ｼﾒﾃﾞｽ｡":PR
```



グラフィック 麻雀ゲーム プログラム・リスト

```
INT"ﾃﾝｽｳ ﾊ ";:COLOR2:PRINT"27000ﾃﾝ";:COLOR7:PRINT" ﾉ ﾓﾁﾃﾝ ﾃﾞ ";:COLOR2:PRINT"300
00ﾃﾝ";:COLOR7:PRINT"  ｶﾞｴｼ ﾃﾞｽ｡":PRINT
130 COLOR2:PRINT"Computer";
131 COLOR7:PRINT" ﾉ ﾂﾖｻ ｦ ﾆｭｳﾘｮｸ ｼﾃｸﾀﾞｻｲ"
135 PRINT"ﾂﾖｲ  Computer....1"
140 PRINT"ﾌﾂｳﾉ Computer....2"
145 INPUT"ﾖﾜｲ  Computer....3";N7
150 N8=10+15*(N7-1):N9=73-N8
210 DATA a,b,c,d,e,f,g,h,i,j,k,l,m,n
220 RESTORE210:FORI=1TO14:READR$(I):NEXT
230 FORI=1TO4:F(I)=27000:NEXT
235 GOSUB3800
250 WIDTH40:COLOR4,255,1:PRINTCHR$(12):COLOR3:PRINT"ｲﾏ ﾊﾟｲ ｦ ﾅﾗﾍﾞﾃ ｲﾏｽ."
260 GOTO1820
270 'ｼﾞﾌﾞﾝ ﾉ ﾘｰﾁ ﾉ ﾛﾝ ﾁｪｯｸ
280 IFC(4,I1)=C(4,15)THENP3=0:GOTO305
290 IFC(4,I1)=C(4,16)THENP3=0:GOTO305
300 GOTO1310
305 J3=4:PRINTCHR$(12):LINE(0,1)-(78,13),"■",4,BF:GOTO940
310 'ﾂﾖｻ
320 A=RND(1)*N9+N8:RETURN
350 IFP(J1)=0THENRETURN
360 IFL<Q(J1)THENRETURN
370 P8(J1)=P8(J1)+1:X=0:Y=5+(J1-1)*4:LOCATEX,Y:COLOR2:PRINT"*ﾘｰﾁ!":FORI=1TO50:BE
EP1:NEXT:BEEP0:RETURN
400 'ﾛﾝ ﾅｶﾞﾚ ﾁｪｯｸ
410 C(J1,14)=E(L):E=E(L):IFP(J1)=0THEN430
420 IFL>Q(J1)-1THEN440
430 A=RND(1)*65+8:IFL<ATHEN460
440 IFC(J1,14)=C(J1,15)THENGOTO7200
450 IFC(J1,14)=C(J1,16)THENGOTO7200
460 I1=14
470 'ｽﾃﾊﾟｲ ﾃﾞｨｽﾌﾟﾚｲ
480 L1=INT((L-1)/4)+1:M(J1,L1)=E:Y1=12+(J1-1)*16:Y=Y1:IFL1>14THENGOSUB3500:GOTO4
90
485 X=L1*10:GOSUB40
490 TIME$="00:00:00"
500 K$=INKEY$:IFK$=" "THEN3090
510 IFTIME$="00:00:02"THENGOTO520ELSE500
520 IFL>72THEN1160
530 IFJ1=1THEN1180
540 IFJ1=2THEN1220
550 GOTO1260
560 'ﾘｰﾁ ｽﾃﾊﾟｲ ﾉ ﾃﾞｨｽﾌﾟﾚｲ
570 C(4,14)=E(L):L1=INT((L-1)/4)+1:Y=60:IFL1>14THENGOSUB3500:RETURN
575 X=L1*10:GOSUB40:RETURN
580 'ﾀﾁｬ ﾉ ﾛﾝ ﾁｪｯｸ
590 GOSUB310:IFP8(J3)>0THEN600
595 IFL<ATHENRETURN
600 IFC(J1,14)=C(J3,15)THENGOSUB7000:GOTO625
610 IFC(J1,14)=C(J3,16)THENGOSUB7000:GOTO625
620 Q1=0:RETURN
625 IFX=0THENQ1=1:RETURNELSEQ1=0:RETURN
630 'a-n ﾃﾞｨｽﾌﾟﾚｲ
640 LOCATE2,22:COLOR5:PRINT"A    B    C    D    E    F    G    H    I    J    K
  L    M    N ";:RETURN
650 'ﾘｰﾁ ﾉ ﾏﾁ ﾉ input ﾄ ｾﾝｹﾞﾝ
660 LOCATE1,23:INPUT"ﾏﾁｰ-1";M$:GOSUB3550:C(4,15)=O3:IFO3=99THEN660
662 GOSUB3510:LOCATE1,23:COLOR7:INPUT"ﾏﾁｰ-2";M$:GOSUB3550:C(4,16)=O3:IFO3=99THEN
662
665 GOSUB3510:LOCATE0,17:COLOR2:PRINT"*ﾘｰﾁ!":FORI=1TO50:BEEP1:NEXT:BEEP0:GOTO145
0
670 'ﾘｰﾊﾟｲ
680 I=1
690 FORJ=I+1TO14-N4*3
700 IFC(4,I)<C(4,J)THEN720
710 SWAPC(4,I),C(4,J)
720 NEXT
730 I=I+1
740 IFI<14-N4*3THEN690
750 RETURN
760 'ﾊﾟｲ ﾉ ｼｭﾂﾘｮｸ
770 J1=4:I1=1
780 X=I1*10-10:Y=76:GOSUB40:I1=I1+1:IFI1<N2+1THEN780
790 'ﾂﾓﾊﾟｲ ｦ ｹｽ
800 LINE(72,19)-(76,21),"■",4,BF:RETURN
810 PRINTCHR$(12):LINE(0,1)-(78,13),"■",4,BF:COLOR7:IFJ1=1THENB$=U$
820 IFJ1=2THENB$=V$
830 IFJ1=3THENB$=W$
840 LOCATE5,0:PRINTB$;" ﾉ ﾂﾓﾛﾝ":P1=J1:GOTO970
850 IFJ1=1THENB$=U$
860 IFJ1=2THENB$=V$
870 IFJ1=3THENB$=W$
880 IFJ1=4THENB$=X$
890 PRINTCHR$(12):LINE(0,1)-(78,13),"■",4,BF:COLOR7:LOCATE5,0:PRINTB$;" ﾉ ﾌﾘｺﾐ
";:P3=J1:IFJ3=1THENC$=U$
900 IFJ3=2THENC$=V$
910 IFJ3=3THENC$=W$
920 IFJ3=4THENC$=X$
```

グラフィック 麻雀ゲーム プログラム・リスト

```
930 PRINTC$;" ﾉ ﾛﾝ":J1=J3:P2=J3:GOTO970
940 IFP3=0THENN2=14:P1=4:GOTO960
950 P2=4:N2=14
960 J1=4:'ｱｶﾞﾘ ﾅｶﾞﾚｼｮﾘ
970 'ｱｶﾞﾘﾊﾟｲ ﾉ ﾃﾞｨｽﾌﾟﾚｲ
980 FORI=1TO100:BEEP1:NEXT:BEEP0
990 IF(P1=4)+(P2=4)THEN1010
1000 N2=13:GOTO1020
1010 N2=13-N4*3
1020 FORI1=1TON2:X=I1*10:Y=8:GOSUB40:NEXTI1
1030 K(1)=E:IFP1=4THENI1=14-N4*3:GOTO1050
1040 C(J1,I1)=E
1050 X=140:Y=8:GOSUB40
1060 GOSUB3130:IFP2=0THEN1080
1070 IFP2<>4THEN1150
1080 IFN4=0THEN1150
1090 IFP1=0THEN1110
1100 IFP1<>4THEN1150
1110 GOSUB3030
1150 GOTO2610
1160 COLOR7:PRINTCHR$(12):PRINT"ﾅｶﾞﾚﾃﾞｽ":Z3=0:GOTO2630
1170 L=L+1:J1=1:GOSUB350:GOTO400:'ｼﾓﾁｬ
1180 J3=2:GOSUB590:IFQ1=1THEN850
1190 J3=3:GOSUB590:IFQ1=1THEN850
1200 J3=4:GOSUB600:IFQ1=1THEN850
1210 L=L+1:J1=2:GOSUB350:GOTO400:'ﾄｲﾒﾝ
1220 J3=3:GOSUB590:IFQ1=1THEN850
1230 J3=4:GOSUB600:IFQ1=1THEN850
1240 J3=1:GOSUB590:IFQ1=1THEN850
1250 L=L+1:J1=3:GOSUB350:GOTO400:'ｶﾐﾁｬ
1260 J3=4:GOSUB600:IFQ1=1THEN850
1270 J3=1:GOSUB590:IFQ1=1THEN850
1280 J3=2:GOSUB590:IFQ1=1THEN850
1290 L=L+1:I1=14-N4*3:J1=4:C(J1,I1)=E(L):E=E(L):LOCATE72,22:PRINT"ﾂﾓﾊｲ":X=144:Y
=76:GOSUB40:'ﾂﾓﾊｲ ﾃﾞｨｽﾌﾟﾚｲ
1300 IFP6=4THENGOTO270
1310 IFP6=4THENP8(4)=P8(4)+1:GOSUB560:GOSUB790:GOTO1430
1320 GOSUB670:GOSUB3140:N2=14-N4*3:GOSUB770:IFN4<>0THENGOSUB3030
1340 LOCATE5,24:COLOR7:PRINT" ﾘｰﾁ･･･'ﾘｰﾁ' ｼｮﾁ･･･Space key      (ﾛﾝ･･'R'  ﾁｲ･･'C'
  ﾎﾟﾝ･･'P')    ";:LOCATE1,23:INPUT"ｽﾃﾊﾟｲ";F$:IFF$="r"THENPRINTCHR$(12):LINE(0,1)-(
78,13),"■",4,BF:P3=0:J3=4:GOTO940
1350 IFF$="ﾘｰﾁ"THENP6=4:P8(4)=1
1360 GOSUB3510:IFP6=4THENLOCATE1,23:COLOR7:INPUT"ｽﾃﾊﾟｲ";F$
1370 GOSUB3510:FORF1=1TO14:IFF$=R$(F1)THEN1390
1380 NEXT
1385 LOCATE1,23:PRINT"ｽﾃﾊﾟｲ ﾊ A-N ﾃﾞｽ.":FORI=0TO1000:NEXT:GOSUB3510:GOTO1340
1390 I1=F1:E=C(4,I1):X=F1*5-5:Y=18:LINE(X,Y)-(X+4,Y+3),"■",4,BF:L1=INT((L-1)/4)+
1:M(4,L1)=E:IFL1>14THENY=60:GOSUB3500:GOTO1397
1395 X=L1*10:Y=60:GOSUB40
1397 FORJ=F1TO13:C(4,J)=C(4,J+1):NEXT:'ｽﾃﾊﾟｲ ｦ ﾂﾒﾙ
1400 N2=13-N4*3
1410 GOSUB3140
1420 GOSUB770:IFP6=4THEN650
1430 IFN4<>0THENGOSUB3030
1450 GOSUB310
1460 J1=4:C(J1,14)=E
1465 FORJ3=1TO3:IFP8(J3)>0THEN1470
1467 IFL<A THEN1475
1470 GOSUB600:IFQ1=1THEN850
1475 NEXTJ3
1480 IFL>72THEN1160
1490 GOTO1170
1500 '
1510 'ｻｲｺﾛ ﾌﾘ
1520 COLOR5:LOCATE3,6:PRINT"ｻｲ  No.1.....";H1:LOCATE3,9:PRINT"ｻｲ  No.2.....";H2
1580 COLOR7:LOCATE4,12:PRINT"ｱﾅﾀ ﾉ ﾊﾞｼｮ ﾊ 「";:COLOR2:PRINTA$;:COLOR7:PRINT"」ﾃﾞｽ
.":FORI=1TO5000:NEXT
1585 WIDTH80,25:COLOR4,255,1
1590 PRINTCHR$(12):L=0:N3=N3+1:IFRE>0THENN3=N3-1
1592 M4=M4+1:IFRE>0THENM4=M4-1
1595 IFN3=5THENN3=1
1600 LOCATE5,1:COLOR5:PRINT"*** 4ﾆﾝ ﾏｰｼﾞｬﾝ ｹﾞｰﾑ ***":LOCATE50,1:COLOR6:PRINT"ﾄﾝ"
;N3:COLOR7:LOCATE35,0:PRINT"ｱﾅﾀ ﾉ ﾊﾞｼｮ":LOCATE40,1:PRINTA$:LOCATE60,1:PRINT"ﾄﾞﾗ
":J1=4:I1=14:C(J1,I1)=E%(82):X=130:Y=0:GOSUB40
1605 COLOR7:LOCATE0,3:PRINTU$:LOCATE0,4:PRINTY$:LOCATE0,7:PRINTV$:LOCATE0,8:PRIN
TY$:LOCATE0,11:PRINTW$:LOCATE0,12:PRINTY$:LOCATE0,15:PRINTX$:LOCATE0,16:PRINTY$
1610 C(4,14)=38:GOSUB670:N2=13:GOSUB770:GOSUB630:IFA$="ﾍﾟｲ"THEN1170
1620 IFA$="ｼｬｰ"THEN1210
1630 IFA$="ﾅﾝ"THEN1250
1640 GOTO1290
1650 G=INT(RND(1)*3):H=G*10+F:T=0
1660 IFD(H+T)=4THENONUGOTO1960,2120,2190
1670 T=T+1
1680 IFT<S+1THEN1660
1690 V=0
1700 D(H+V)=D(H+V)+1:V=V+1:IFV<S+1THEN1700
1710 RETURN
1720 H=INT(RND(1)*37)+1:IFH/10=INT(H/10)THEN1720
1730 IFD(H)<PTHEN1750
```

```
1740 GOTO1720
1750 D(H)=D(H)+Q:RETURN
1820 'ﾒﾓﾘ ﾉ ｸﾘﾔ
1830 P1=0:P2=0:P3=0:P4=0:P6=0:P7=0:R1=0:S1=0:N4=0:N5=0:Q1=0:Z1=0:FORI=1TO4
1840 FORJ=1TO16:C(I,J)=0:NEXTJ:NEXTI
1845 FORI=1TO18:K(I)=0:NEXT
1850 FORI=1TO37:D(I)=0:L(I)=0:NEXT
1855 FORI=1TO4:FORJ=1TO21:M(I,J)=0:NEXTJ:NEXTI
1860 FORI=1TO84:E(I)=0:NEXT
1865 FORI=1TO4:P8(I)=0:NEXT
1870 'ﾊｲﾊﾟｲ
1880 J1=0
1890 GOSUB3520:J1=J1+1:L=INT(RND(1)*6):IFL=5THEN2350
1900 IFL=4THENA=4:GOTO1920
1910 A=3
1920 B=INT(RND(1)*A+1):IFB=0THEN2050
1930 'ｼﾞｭﾝﾂ
1940 GOSUB3520:E1=0
1950 E1=E1+1
1960 F=INT(RND(1)*7)+1:S=2:U=1
1970 GOSUB1650
1980 C(J1,(E1-1)*3+1)=H:C(J1,(E1-1)*3+2)=H+1:C(J1,(E1-1)*3+3)=H+2:IFE1<BTHEN1950
1990 IFA=3THEN2010
2000 GOTO2030
2010 IFB=3THEN2070
2020 GOTO2050
2030 IFB=4THEN2070
2040 'ﾄｲﾂ
2050 GOSUB3520:FORI=1TOA-B:P=2:Q=3:GOSUB1720:C(J1,(I-1+B)*3+1)=H:C(J1,(I-1+B)*3+
2)=H:C(J1,(I-1+B)*3+3)=H:NEXT
2060 'ｱﾀﾏ
2070 IFL=4THEN2370
2080 P=3:Q=2:GOSUB1720:C(J1,10)=H:C(J1,11)=H
2090 'ﾘｬﾝﾒﾝ
2100 IFL=0THEN2120
2110 GOTO2170
2120 F=INT(RND(1)*6+2):S=1:U=2
2130 GOSUB1650
2140 C(J1,12)=H:C(J1,13)=H+1:C(J1,15)=H-1:C(J1,16)=H+2:IFJ1<3THEN1890
2150 GOTO2420
2160 'ﾍﾟﾝﾁｬﾝ
2170 IFL=1THEN2190
2180 GOTO2280
2190 N=INT(RND(1)*2):IFN=0THENF=8:GOTO2210
2200 F=1
2210 S=1:U=3
2220 GOSUB1650
2230 C(J1,12)=H:C(J1,13)=H+1:IFF=1THENC(J1,15)=H+2:GOTO2250
2240 C(J1,15)=H-1
2250 C(J1,16)=0:IFJ1<3THEN1890
2260 GOTO2420
2270 'ｶﾝﾁｬﾝ
2280 GOSUB3520:IFL=2THEN2300
2290 GOTO2410
2300 F=INT(RND(1)*7)+1:G=INT(RND(1)*3):H=G*10+F:IFD(H)=4THEN2300
2310 IFD(H+2)=4THEN2300
2320 D(H)=D(H)+1:D(H+2)=D(H+2)+1:C(J1,12)=H:C(J1,13)=H+2:C(J1,15)=H+1:C(J1,16)=0
:IFJ1<3THEN1890
2330 GOTO2420
2340 'ﾁｰﾄｲﾂ
2350 GOSUB3520:FORM=1TO6:P=3:Q=2:GOSUB1720:C(J1,1+(M-1)*2)=H:C(J1,2+(M-1)*2)=H:N
EXT
2360 'ﾀﾝｷ
2370 P=4:Q=1:GOSUB1720:C(J1,13)=H
2380 C(J1,15)=H:C(J1,16)=0:IFJ1<3THEN1890
2390 GOTO2420
2400 'ｼｬﾝﾎﾟﾝ
2410 P=3:Q=2:GOSUB1720:C(J1,12)=H:C(J1,13)=H:C(J1,15)=H:C(J1,16)=C(J1,10):IFJ1<3
THEN1890
2420 '
2430 GOSUB3520:FORI=1TO13
2440 H=INT(RND(1)*37)+1:IFH/10=INT(H/10)THEN2440
2450 IFD(H)=4THEN2440
2460 C(4,I)=H:D(H)=D(H)+1:NEXT
2470 'd(i) ｦ e(l) ﾆｲﾚﾙ
2480 L=0
2490 FORI=1TO37:D(I)=4-D(I):NEXT
2500 FORI=1TO37:IFI/10=INT(I/10)THEN2530
2510 FORK=1TOD(I):IFD(I)=0THEN2530
2520 L=L+1:E(L)=I:NEXTK
2530 NEXTI
2540 'e(l) ｦｷﾙ
2550 PRINT:PRINT"ｺﾚｶﾗ ｻｲ ｦ ﾌﾘﾏｽ."
2560 FORI=1TO200:A=INT(RND(1)*42)+1:B=INT(RND(1)*42)+43:SWAPE(A),E(B):NEXT
2570 'ﾘｰﾁ ｾﾝｹﾞﾝﾃｲｷﾞ
2580 FORI=1TO3:P(I)=INT(RND(1)*2):NEXT
2590 FORI=1TO3:Q(I)=RND(1)*N9+N8:NEXT
2600 PRINTCHR$(12):GOTO1500
2610 GOSUB4410
```

グラフィック 麻雀ゲーム　プログラム・リスト

```
2620 GOSUB5500
2630 FORI=1TO3:IFP8(I)>0THENFR=FR+1000:F(I)=F(I)-1000
2640 NEXT
2650 IFP6>0THENFR=FR+1000:F(4)=F(4)-1000
2660 IFP1>0THENF(P1)=F(P1)+FR:FR=0
2670 IFP2>0THENF(P2)=F(P2)+FR:FR=0
2910 LOCATE45,15:COLOR6:PRINT"** ﾃﾝｽｳ ﾋｮｳ **":LOCATE45,16:COLOR7:PRINT" ( ｱｶﾞﾘﾃﾝ
";Z3;")"
2915 LOCATE40,17:COLOR4:PRINTU$,:COLOR7:PRINTF(1)
2916 LOCATE40,19:COLOR4:PRINTV$,:COLOR7:PRINTF(2)
2917 LOCATE40,21:COLOR4:PRINTW$,:COLOR7:PRINTF(3)
2918 LOCATE40,23:LINE23,1:COLOR5:PRINTX$,F(4)
2919 IFM4>7THEN6200:'ﾊﾝﾁｬﾝ ｵﾜﾘ
2920 COLOR7:PRINT"ﾏﾀ ｻｲ ｦ ﾌｯﾃ ｸﾀﾞｻｲ.Push any key";
2921 IFINKEY$=""THEN2921
2922 PRINTCHR$(12)
2925 IFP5=P1ORP5=P2THENRE=RE+1:GOTO250ELSERE=0:GOSUB3710:GOTO250
2930 'ﾜｲﾊｲ
2940 N4=N4+1:LINE(X,Y)-(X+9,Y+11),PSET,4,BF:I=1+(N4-1)*3:G(I)=E(L):L=L+1:COLOR7:
LOCATE1,23:INPUT"ﾜｲﾊｲ-1 ﾊ? (A-N) ";F$:K=2+(N4-1)*3:GOSUB3000:LOCATE1,23:INPUT"ﾜｲ
ﾊｲ-2 ﾊ? (A-N) ";F$:K=3+(N4-1)*3:GOSUB3000:N5=N5+1:GOTO1340
2950 I2=F1
2960 C(4,I2)=C(4,I2+1):I2=I2+1
2970 IFI2<14THEN2960
2980 N5=N5+1:N2=13-N5:GOSUB3140:GOSUB770:RETURN
3000 LOCATE1,23:COLOR4:PRINT"[redacted]";:F
ORF1=1TO14:IFF$=R$(F1)THEN3020
3010 NEXT
3020 J1=4:I1=F1:G(K)=C(J1,I1):C(J1,I1)=38:GOSUB2950:RETURN
3030 X=100-(N4-1)*30:FORI=1TON4*3-1
3040 FORJ=1+ITON4*3
3050 IFG(I)<G(J)THEN3070
3060 SWAPG(I),G(J)
3070 NEXTJ:NEXTI
3080 FORI=1TON4*3:J1=4:I1=11+I-(N4-1)*3:C(J1,I1)=G(I):X=X+10:Y=76:IFP1>0ORP2>0TH
ENY=24
3085 GOSUB40:LOCATEX/2,Y/4+3:COLOR6:PRINT" ﾅｷ  ":NEXT:RETURN
3090 LOCATE1,23:INPUT"ｼｮﾁ";C$:LOCATE1,23:COLOR4:PRINT"[redacted]";:IFC$="p"THENS
1=S1+1:GOTO2930
3095 IFC$="c"THENR1=R1+1:IFJ1<>3THENP2=4:J1=4:X=1:M=0:N=0:PRINTCHR$(12):GOTO2620
ELSEGOTO2930
3100 IFC$="r"THENP1=0:P2=4:P3=J1:J1=4:J3=4:PRINTCHR$(12):LINE(0,1)-(78,13),"■",4
,BF:GOTO970
3105 GOTO3090
3130 LOCATE2,12:PRINT"ﾄﾞﾗﾊﾟｲ":I1=0:C(J1,I1)=E(82):X=20:Y=40:GOSUB40:LOCATE18,12:
PRINT"ｳﾗﾄﾞﾗ":C(J1,I1)=E(83):X=52:Y=40:GOSUB40:RETURN
3140 LINE(0,19)-(77,21),"■",4,BF:RETURN
3180 LINE(0,20)-(77,21),"■",4,BF:RETURN
3500 FORI=20TO140STEP10:GET@(I,Y)-(I+9,Y+11),B%,G:COLOR7:PUT@(I-10,Y)-(I-1,Y+11)
,B%,PSET:NEXTI:X=140:GOSUB40:RETURN
3510 LOCATE1,23:COLOR4:PRINT"[redacted]";:RETURN
3520 W=INT(RND(1)*38):IFINT(W/10)=W/10THEN3520
3525 X=INT(RND(1)*32)*2:Y=INT(RND(1)*18)*4+12
3530 ONWGOTO41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
,64,65,66,67,68,69,70,71,72,73,74,75,76,77
3550 O4=ASC(M$)
3555 IF(O4>57)+(O4<48)THEN3600
3560 O3=0:O$=MID$(M$,2,1):IF(O$="ﾋ")+(O$="ﾏ")+(O$="ｿ")THEN3570ELSEGOTO3670
3570 IFO$="ﾏ"THENO3=10
3580 IFO$="ｿ"THENO3=20
3590 O3=O3+O4-48:RETURN
3600 IFLEFT$(M$,1)="ﾄ"THENO3=31:RETURN
3610 IFLEFT$(M$,1)="ﾅ"THENO3=32:RETURN
3620 IFLEFT$(M$,1)="ｼ"THENO3=33:RETURN
3630 IFLEFT$(M$,1)="ﾍ"THENO3=34:RETURN
3640 IFM$="ﾊﾂ"THENO3=35:RETURN
3650 IFLEFT$(M$,1)="ﾁ"THENO3=36:RETURN
3660 IFM$="ﾊｸ"THENO3=37:RETURN
3670 O3=99:RETURN
3700 'ﾊﾞｼｮ ｦ ｽｽﾒﾙ
3710 P5=P5+1:IFP5>4THENP5=1
3720 IFP5=1THENA$="ﾍﾟｲ":GOTO3760
3730 IFP5=2THENA$="ｼｬｰ":GOTO3760
3740 IFP5=3THENA$="ﾅﾝ":GOTO3760
3750 IFP5=4THENA$="ﾄﾝ"
3760 RETURN
3800 H1=INT(RND(1)*6)+1:H2=INT(RND(1)*6)+1:H3=H1+H2
3805 FORI=0TO2:P5=H3-1-I*4:IFP5<5THEN3810ELSENEXTI
3810 FORA=0TO2:IFH3=2+A*4THENA$="ﾍﾟｲ":H4=4:GOTO3860
3820 IFH3=3+A*4THENA$="ｼｬｰ":H4=3:GOTO3860
3830 IFH3=4+A*4THENA$="ﾅﾝ":H4=2:GOTO3860
3840 IFH3=5+A*4THENA$="ﾄﾝ":H4=1:GOTO3860
3850 NEXT
3860 RETURN
4400 'ﾔｸ ﾉ ｹｲｻﾝ M=ﾔｸﾏﾝ N=ﾔｸｽｳ
4410 LOCATE5,15:PRINT"** ﾔｸ **"
4415 IFJ1<>4THENN4=0:S1=0:R1=0
4420 GOSUB7005:IFX=1THENPRINT"ﾌﾘﾃﾝ";:GOTO4750
```

```
4430 N=0:M=0:Z1=0:SU=0:X=14:GOSUB6000:IFN4>0THEN4465
4440 IFP8(J1)>0THENPRINT"ﾘｰﾁ･･･････････ 1":N=N+1
4450 IFP8(J1)=1THENPRINT"ｲｯﾊﾟﾂ･････････ 1":N=N+1
4460 IFP1>0THENPRINT"ﾂﾓ･･･････････ 1":N=N+1
4465 H5=5-P5+J1:IFH5>4THENH5=H5-4
4470 FORI=1TO13STEP2:IFE(I)<>E(I+1)THEN4650
4480 NEXTI:GOSUB8010
4490 IFE(1)>30ORFNA(E(1))<>FNA(E(14))THEN4510ELSEPRINT"ﾀﾞｲｼｬﾘﾝ･･････ ﾔｸﾏﾝ":M=M+
1:GOTO5470
4510 IFE(14)>28THEN4550
4520 FORI=1TO13STEP2:P=FNB(E(I)):IFP=1ORP=9THEN4550
4540 NEXTI:PRINT"ﾀﾝﾔｵ･････････ 1":N=N+1:GOTO4640
4550 IFE(1)<30THEN4570
4560 PRINT"ﾂｳｲｰｿｳ ﾁｰﾄｲﾂ･･ ﾔｸﾏﾝ":M=M+1:GOTO5470
4570 FORI=1TO13STEP2:IFE(I)>30THEN4590
4580 P=FNB(E(I)):IFP>1ANDP<9THEN4600
4590 NEXTI:PRINT"ﾎﾝﾛｰﾄｳ･･･････ 2":N=N+2:GOTO4640
4600 P=FNA(E(1)):FORI=3TO13STEP2
4610 IFE(I)>30THEN4630
4620 IFP<>FNA(E(I))THEN4640
4630 NEXTI:PRINT"ｼｮｰ ｼｬﾘﾝ･････ 6":N=N+6:GOTO5450
4640 PRINT"ﾁｰﾄｲﾂ････････ 2":Z1=100:N=N+2:GOTO5450
4650 P=0:FORI=2TO14:IFE(I-1)=E(I)THENP=P+1:Q=E(I)
4660 NEXTI:IFP<>1THEN4710
4670 FORI=1TO14:IFE(I)>30THEN4690
4680 P=FNB(E(I)):IFP>1ANDP<9THEN4710
4690 NEXTI:PRINT"ｺｸｼﾑｿｳ･･･････ ﾔｸﾏﾝ":M=M+1:IFQ<>K(1)THEN5470
4700 PRINT"13ﾒﾝ ﾏﾁ･･････ ﾔｸﾏﾝ":M=M+1:GOTO5470
4710 FORI=1TO14:P=4+FNA(E(I)):K(P)=K(P)+1:NEXTI
4720 P=0:Q=0:FORI=4TO7:IFK(I)/3=INT(K(I)/3)THENP=P+1
4730 IF(K(I)+1)/3=INT((K(I)+1)/3)THENQ=Q+1:K(8)=I
4740 NEXTI:IFP=3ANDQ=1THEN4760
4750 LINE18,1:LOCATE2,18:PRINT"ﾁｮﾝﾎﾞ!!  8000ﾃﾝ ｹﾞﾝﾃﾝ":N=0:M=0:RETURN
4760 P=(K(8)-4)*10+1:Q=P+8:IFK(8)=7THENQ=37
4770 GOSUB8000:FORI=PTOQ:IFL(I)<2THEN4810
4790 L(I)=L(I)-2:K(18)=I:GOSUB9000:IFX=0THEN4820
4800 GOSUB8000
4810 NEXTI:GOTO4750
4820 FORJ=4TO7:IFJ=K(8)THEN4850
4830 P=(J-4)*10+1:Q=P+8:IFJ=7THENQ=37
4840 GOSUB9000:IFX=1THEN4750
4850 NEXTJ:GOSUB8010
4860 GOSUB6100:ONS-13GOTO5120,5120,4970,4870,5210
4865 IFS2=1THEN4875ELSE4880
4870 IFS2<>0THEN4880
4875 PRINT"ｻﾝｱﾝｺｳ･･･････ 2":N=N+2
4880 FORX=1TO4:X1=K(13+X):IFX1>30THEN4882
4881 A(X)=FNB(X1)
4882 NEXT
4883 FORX= 1TO3:FORY=1TO4-X:IFA(Y)<=A(Y+1)THEN4886
4884 SWAPA(Y),A(Y+1)
4886 NEXTY:NEXTX
4890 IF(A(1)=A(3)ANDA(1)>0)OR(A(2)=A(4)ANDA(2)>0)THENPRINT"ｻﾝｼｮｸ ﾄﾞｳｺｳ･･･ 2":N=N
+2:GOTO5310ELSE4920
4920 IF(A(1)+2=A(3)ANDA(1)>0)OR(A(2)+2=A(4)ANDA(2)>0)THENPRINT"ｻﾝﾚﾝｺｳ･･･････ 2"
:N=N+2:GOTO4980
4930 IFL(31)+L(32)+L(33)+L(34)<>11THEN4950
4940 PRINT"ｼｮｳｽｳｼ･･･････ ﾔｸﾏﾝ":M=M+1:GOTO5470
4950 IFL(35)+L(36)+L(37)=9THENPRINT"ﾀﾞｲｻﾝｹﾞﾝ･････ ﾔｸﾏﾝ":M=M+1:GOTO5470
4960 IFSU=1THEN5030
4970 IFL(35)+L(36)+L(37)=8THENPRINT"ｼｮｳｻﾝｹﾞﾝ･････ 2":N=N+2:GOTO5100
4980 IFFNA(E(1))<>FNA(E(14))THEN5100
4990 Y=6:IFN4>0THENY=5
4995 PRINT"ﾁﾝｲﾂ･･････････";Y:N=N+Y
5000 FORI=3TO11:IFE(I)+1<E(I+1)THEN5030
5010 NEXTI:IFE(3)+8<>E(12)THEN5030
5020 PRINT"ﾁｭｳﾚﾝﾎﾟｳﾄｳ･･･ ﾀﾞﾌﾞﾙ":M=M+2:GOTO5470
5030 FORI=1TO14:IFE(I)=22ORE(I)=23ORE(I)=24THEN5070
5040 IFE(I)=26ORE(I)=28ORE(I)=35THEN5070
5050 GOTO5080
5070 NEXTI:PRINT"ﾘｭｳｲｰｿｳ･･････ ﾀﾞﾌﾞﾙ":M=M+2:GOTO5470
5080 IFSU=1THEN5470
5090 GOTO5310
5100 FORI=4TO6
5105 IFK(I)+K(7)<>14THEN5110
5107 Y=3:IFN4>0THENY=2
5108 PRINT"ﾎﾝｲﾂ･･････････";Y:N=N+Y:GOTO5030
5110 NEXTI:GOTO5310
5120 X1=C(J1,15):X2=C(J1,16):IFX1>30ORX2>30ORFNA(X1)<>FNA(X2)ORABS(X1-X2)<>3THEN
5135
5125 IFP1>0ORN4>0ORR-10<>4ORK(18)=31ORK(18)>34ORK(18)-30=H5THEN5135
5130 PRINT"ﾋﾟﾝﾌ･････････ 1":N=N+1
5135 P=1:FORI=10TO12:FORJ=I+1TO13
5140 IFK(I)<>K(J)ANDFNB(K(I))=FNB(K(J))THENP=P+1
5150 NEXTJ,I:Y=2:IFN4>0THENY=1
5160 IFP>3THENPRINT"ｻﾝｼｷ･･････････";Y:N=N+Y:GOTO5310
5165 IFK(10)+3=K(11)ANDK(11)+3=K(12)THEN5200
5170 IFK(10)+3=K(11)ANDK(11)+3=K(13)THEN5200
5175 IFK(10)+3=K(12)ANDK(12)+3=K(13)THEN5200
```

```
5180 IFK(11)+3=K(12)ANDK(12)+3=K(13)THEN5200
5190 GOTO4980
5200 Y=2:IFN4>0THENY=1
5205 PRINT"ｲｯｷﾂｳｶﾝ･･･････";Y:N=N+Y:GOTO4980
5210 IFS2>0THEN5230ELSEPRINT"ｽｰｱﾝｺｳ････････ ﾔｸﾏﾝ":M=M+1:SU=1
5220 IFK(1)=K(18)THENPRINT"ﾀﾝｷ･･･････････ ﾔｸﾏﾝ":M=M+1
5230 IFL(1)+L(9)+L(11)+L(19)+L(21)+L(29)<>14THEN5250
5240 PRINT"ﾁﾝﾛｰﾄｳ････････ ﾔｸﾏﾝ":M=M+1:GOTO5470
5250 IFK(14)+3<>K(17)ORK(17)>30ORK(14)-10*INT(K(14)/10)>6THEN5280
5270 PRINT"ｽｰﾚﾝｺｳ････････ ﾔｸﾏﾝ":M=M+1:GOTO5470
5280 IFE(1)>30THENPRINT"ﾂｳｲｰｿｳ････････ ﾔｸﾏﾝ":M=M+1
5290 IFL(31)+L(32)+L(33)+L(34)<>12THEN5305
5300 PRINT"ﾀﾞｲｽｳｼ････････ ﾀﾞﾌﾞﾙ":M=M+2:GOTO5470
5305 IFSU<>1THENPRINT"ﾄｲﾄｲ･･････････ 2":N=N+2:GOTO4865
5307 GOTO4930
5310 P=0:Q=0:FORI=14TO18:IFK(I)>30THENQ=Q+1:GOTO5330
5320 R=FNB(K(I)):IFR=1ORR=9THENP=P+1
5330 NEXTI:FORI=10TO13:R=FNB(K(I))
5340 IFR=1ORR=7THENP=P+1
5350 NEXTI:IFP=0ANDQ=0THENPRINT"ﾀﾝﾔｵ･･････････ 1":N=N+1:GOTO5380
5355 Y=3:IFN4>0THENY=2
5360 IFP=5THENPRINT"ｼﾞｭﾝﾁｬﾝﾀ･････";Y:N=N+Y:GOTO5380
5365 Y=2:IFN4>0THENY=1
5370 IFP+Q=5THENPRINT"ﾁｬﾝﾀ･･････････";Y:N=N+Y
5380 IFN4>0THEN5400
5385 FORI=1OTO26-S:IFK(I)=K(I+1)ANDK(I)>0THENPRINT"ｲｲﾍﾟｲｺｳ······ 1":N=N+1
5390 NEXTI
5400 IFL(35)=3THENPRINT"ﾊﾂ･･････････････ 1":N=N+1
5410 IFL(36)=3THENPRINT"ﾁｭﾝ･････････････ 1":N=N+1
5420 IFL(37)=3THENPRINT"ﾊｸ･･････････････ 1":N=N+1
5430 IFL(31)=3THENPRINT"ﾄﾝ･･････････････ 1":N=N+1
5440 IFH5=1ANDL(31)=3THENPRINT"ﾄﾝ･･･････････ 1":N=N+1
5444 IFH5=2ANDL(32)=3THENPRINT"ﾅﾝ･･･････････ 1":N=N+1
5446 IFH5=3ANDL(33)=3THENPRINT"ｼｬｰ･･････････ 1":N=N+1
5448 IFH5=4ANDL(34)=3THENPRINT"ﾍﾟｲ･･････････ 1":N=N+1
5450 IFL>68THENPRINT"ﾊｲﾃｲ･･･････････ 1":N=N+1
5455 IFL<5THENPRINT"ﾃﾝﾎｳ･･････････ ﾀﾞﾌﾞﾙ":M=M+2
5457 IFN=0THENRETURN
5460 J=E(82):IFL(J)>0THENPRINT"ﾄﾞﾗ･･･････････";L(J):N=N+L(J)
5470 J=E(83):IFP8(J1)>0ANDL(J)>0THENPRINT"ｳﾗﾄﾞﾗ･････････";L(J):N=N+L(J)
5480 RETURN
5500 'ｷｿﾃﾝ=a1
5502 IFM>0THENA1=8000*M:A2=A1*2:GOTO5620
5505 IFN>12THENA1=8000:A2=A1*2:GOTO5620
5510 IFN>10THENA1=6000:A2=A1*2:GOTO5620
5515 IFN>7THENA1=4000:A2=A1*2:GOTO5620
5520 IFN>5THENA1=3000:A2=A1*2:GOTO5620
5525 IFN>3THENA1=2000:A2=A1*2:GOTO5620
5527 IFN=0ANDM=0THENGOSUB4750:GOTO5750
5530 IFZ1>99THENZ1=30:GOTO5610
5534 'ﾌ ﾉ ｹｲｻﾝ
5535 Z1=30:IFP7=1THENZ1=22 'ﾂﾓ
5537 IFN4>0THENZ1=20 'ﾅｷ
5540 FORI=14TO17:IFK(I)=0THEN5560
5550 J=FNB(K(I)):IFJ=1ORJ=9ORK(I)>30THENZ1=Z1+8ELSEZ1=Z1+4
5560 NEXT
5570 IFS1>0THENZ1=Z1-S1*4
5580 IFK(18)=31ORK(18)>34ORK(18)-30=H5THENZ1=Z1+2
5590 IFZ1/10=INT(Z1/10)THENGOTO5610
5600 Z1=INT(Z1/10)*10+10
5605 'ﾃﾝ ﾉ ｹｲｻﾝ
5610 Z3=Z1:A1=Z3*2^(N+2):A2=A1+A1
5620 IFP1=0THEN5720:'ﾂﾓﾛﾝ ﾃﾞ ﾅｹﾚﾊﾞ 5720
5625 GOSUB5900
5630 IFP5<>P1THEN5680
5640 'ｵﾔ ﾉ ﾂﾓ
5650 Z3=A2*3:FORI=1TO4:IFI=J1THEN5670
5660 F(I)=F(I)-A2
5670 NEXT:F(J1)=F(J1)+Z3:RETURN
5675 'ｺ ﾉ ﾂﾓ
5680 Z3=A1*2+A2:J2=P5:FORI=1TO4:IFI=J1THEN5710
5690 IFI=J2THENF(I)=F(I)-A2:GOTO5710
5700 F(I)=F(I)-A1
5710 NEXT:F(J1)=F(J1)+Z3:RETURN
5715 'ﾌﾘｺﾐ
5720 J1=P2:J2=P3:A1=A1*4:IFP5=P2THENA1=A1*1.5
5725 GOSUB5900
5730 F(J1)=F(J1)+A1:F(J2)=F(J2)-A1:Z3=A1
5740 RETURN
5749 '5750-5830 ﾂｲｶ
5750 IFP1=P5ORP2=P5THEN5800
5760 Z3=-8000:F(J1)=F(J1)+Z3:F(P5)=F(P5)+4000
5770 FORI=1TO4:IFI=J1ORI=P5THEN5790
5780 F(I)=F(I)+2000
5790 NEXT:RETURN
5800 Z3=-12000:F(J1)=F(J1)+Z3
5810 FORI=1TO4:IFI=J1THEN5830
5820 F(I)=F(I)+4000
5830 NEXT:RETURN
```

グラフィック 麻雀ゲーム　プログラム・リスト

```
5900 'テンスウ ノ キリアゲ
5910 IFA1=INT(AI/100)THEN5920ELSEA1=INT(A1/100)*100+100
5920 IFA2=INT(A2/100)*100THENRETURNELSEA2=INT(A2/100)*100+100:RETURN
6000 FORI=1TO16:E(I)=C(J1,I):NEXT
6010 FORI=1TOX-1:FORJ=1TOX-I
6020 IFE(J)<=E(J+1)THEN6030ELSESWAPE(J),E(J+1)
6030 NEXTJ:NEXTI:RETURN
6100 S2=0:IFP1>0THEN6120
6110 IFK(1)=K(14)ORK(1)=K(15)ORK(1)=K(16)ORK(1)=K(17)THENS2=1
6120 S2=S2+S1:RETURN
6200 FORI=1TO1000:NEXTI
6205 FA=0:FM=0:FORI=1TO4:IFFA<F(I)THENFA=F(I)
6210 NEXT
6220 FORI=1TO4:IFFA=F(I)THENJ=I:GOTO6240
6230 FM=FM+F(I)
6240 NEXT
6250 F(J)=120000!-FM
6260 PRINTCHR$(12):WIDTH36,20:LOCATE10,5:COLOR6:PRINT"** テンスウ ヒョウ **"
6270 LOCATE7,7:COLOR4:PRINTU$,:COLOR7:PRINTF(1)
6280 LOCATE7,9:COLOR4:PRINTV$,:COLOR7:PRINTF(2)
6290 LOCATE7,11:COLOR4:PRINTW$,:COLOR7:PRINTF(3)
6300 LOCATE7,13:LINE13,1:COLOR5:PRINTX$,F(4)
6310 LOCATE3,15:PRINT"ハンチャン オワリ デス。ゴクロウサマデシタ。":END
7000 X=0:IFJ3=4THENRETURN
7005 FORI=1TO21:X1=M(J3,I):IFX1=0THEN7030
7010 IFX1=C(J3,15)ORX1=C(J3,16)ORX1=K(1)THENX=1:RETURN
7020 NEXT
7030 X=0:RETURN
7200 J3=J1:GOSUB7000:IFX=0THENP7=1:GOTO810ELSEGOTO460
8000 R=10:S=14:FORY=10TO17:K(Y)=0:NEXTY
8010 FORY=1TO37:L(Y)=0:NEXTY
8020 FORY=1TO14:L(E(Y))=L(E(Y))+1:NEXTY:RETURN
8500 CONSOLE,,0,1:WIDTH80,25:COLOR6,0,0:PRINTCHR$(12)
8510 A=4:LINE(0,2)-(6,2),"■",A:LINE(6,3)-(6,4),"■",A:LINE(5,5)-(2,8),"■",A:LOCAT
E1,8:COLORA:PRINT"■"
8520 LINE(9,2)-(9,7),"■",A:LOCATE8,8:COLORA:PRINT"■":LINE(12,2)-(12,8),"■",A:LIN
E(13,8)-(14,7),"■",A:LOCATE14,6:COLORA:PRINT"■"
8530 A=6:LINE(18,2)-(16,4),"■",A:LINE(19,2)-(21,2),"■",A:LINE(24,1)-(17,8),"■",A
:LOCATE16,8:COLORA:PRINT"■":LOCATE22,1:PRINT"■":LOCATE25,2:PRINT"■"
8540 LINE(27,2)-(29,2),"■",A:LINE(24,4)-(30,4),"■",A:LINE(30,5)-(27,8),"■",A
8550 LINE(32,2)-(38,2),"■",A:LINE(38,3)-(38,4),"■",A:LINE(37,5)-(34,8),"■",A:LOC
ATE33,8:COLORA:PRINT"■"
8560 LINE(45,3)-(41,7),"■",A:LINE(43,6)-(43,8),"■",A
8570 LINE(49,4)-(49,5),"■",A:LINE(51,4)-(51,5),"■",A:LINE(53,4)-(53,7),"■",A:LIN
E(51,8)-(52,8),"■",A
8580 LINE(58,2)-(56,4),"■",A:LINE(59,2)-(62,2),"■",A:LINE(62,3)-(57,8),"■",A:LOC
ATE56,8:COLORA:PRINT"■"
8590 A=2:LINE(8,11)-(14,11),"■",A:LINE(14,12)-(11,15),"■",A:LINE(9,15)-(11,17),"
■",A
8600 LINE(17,14)-(21,14),"■",A
8610 LINE(24,12)-(26,12),"■",A:LINE(24,14)-(26,14),"■",A:LINE(24,17)-(27,17),"■"
,A:LINE(28,16)-(30,14),"■",A:LINE(29,11)-(30,12),"■",A:LINE(31,11)-(32,12),"■",A
8620 LINE(33,14)-(37,14),"■",A:LINE(37,15)-(36,16),"■",A:LINE(34,13)-(34,17),"■"
,A
8630 LINE(40,12)-(42,12),"■",A:LINE(40,17)-(43,17),"■",A:LINE(44,16)-(46,14),"■"
,A:LOCATE46,13:COLORA:PRINT"■"
8640 A=6:LINE(57,11)-(56,13),"■",A:LINE(58,12)-(62,12),"■",A:LINE(60,13)-(60,16)
,"■",A:LOCATE59,17:COLORA:PRINT"■":LINE(64,11)-(65,12),"■",A:LINE(66,11)-(67,12)
,"■",A
8650 LINE(65,14)-(69,14),"■",A
8660 LINE(76,11)-(72,15),"■",A:LINE(72,16)-(76,16),"■",A:LINE(76,15)-(78,17),"■"
,A
8670 J1=1:I1=1:C(1,1)=INT(RND(1)*38):IFC(1,1)/10=INT(C(1,1)/10)THEN8670
8680 X=98:Y=52:GOSUB40
8700 DATA 1056,0,0,0,0,0,0,0,0,15600,-3186,4408,17648,896,20092,14564,5009,20793
,17684,-19375,20737,17412,4352,20773,4372,5211,20805,17680,4117,16425,4100,4433,
4357,17745,5071,20229,276,271,68,3345
8710 DATA -3183,20753,5140,22353,4476,4368,1984,24336,16881,4369,16765,25874,519
3,4353,17409,4355,20773,4372,5201,-29116,17685,-4079,16400,4148,5201,-28359,1768
3,0
8720 RESTORE8670:FORI=0TO67:READE(I):NEXT
8730 COLOR7:PUT@(20,76)-(151,83),E,PSET
8740 LINE19,1:LINE20,1
8750 DATA 768,0,0,0,0,0,0,-3121,24377,32244,-4081,-32700,14595,5201,-16060,1092,
4113,69,17665,5201,16644,1093,4113,41,17537,-3121,20280,15429,-4079,16,17473,441
7,16704,1093,4113,17,17441,4673,16708,1094,4113,17,17425,-3007,24377,31812,-4081
,-32752,14355,
8760 FORI=0TO49:READE(I):NEXT
8770 PUT@(56,88)-(151,95),E,PSET
8780 LINE22,1:LINE23,1
8790 FORI=1TO2000:A=RND(1):NEXT
8800 RETURN
9000 FORY=PTOQ:IFL(Y)<1THEN9050
9010 IFL(Y)=3THEN9040
9020 IFY>30ORY>Q-2THENX=1:RETURN
9030 L(Y)=L(Y)-1:L(Y+1)=L(Y+1)-1:L(Y+2)=L(Y+2)-1:K(R)=Y:R=R+1:GOTO9000
9040 L(Y)=L(Y)-3:K(S)=Y:S=S+1:GOTO9000
9050 NEXTY
9060 FORY=PTOQ:IFL(Y)<0THENX=1:RETURN
9070 NEXTY:X=0:RETURN
```



PC-8001＋PCG-8100

PCG ギャラクシアン

芸夢狂人

ＨＡＬ研究所から発売されているＰＣＧ-8100は，80字のとき640×200のドットが使え，さらに音も出るなど，すばらしいものです．特にゲームなどに使うと，ゲーム・センターにある本物そっくりにできるし，ＰＣＧとグラフィック・プリンタを組み合わせれば，漢字システムが可能になります．

価格は49,800円と，ちょっと高め(？)ですが，それだけの価値はあると思います(宣伝をしてしまった)．ただ，付属のマニュアルは10ページのペラッとしたもので,これではどうやって使うのか,なかなかわかりません（そのうちリファレンス・マニュアルが発売されるそうですが……)．

そこで，第１章ではいままでＰＣＧを使ってわかったことについて，第２章におまけ（本題かな？）としてＰＣＧを使ったギャラクシアンを載せておきます．

第1章PCGの使い方

GALAXIAN 1 データ転送

PCGAIDを使ってのパターンの作り方はマニュアルを見ればわかると思うので，ここでは省略します．しかし，ＰCGAIDでデータ・テープを作っておくと，ゲームをやるときには，まずデータ・テープを入れてＲＵＮし，次にマシン語を入れ，さらにBASICを入れるというように３回もロードしなくてはなりません．そこで，せめて２回で済むようにマシン語でデータを転送する方法を紹介します．

リスト１はデータ転送プログラムのアセンブル・リストです．これはマニュアルの後に載っているものと同じです．このプログラムはリロケータブル，つまり絶対番地を使っていないので，メモリ上の自分の好きなところに置けます．ギャラクシアンではＤ510番地から始まっています．Ｄ512番地の□で囲んだ２つの数値がデータの先頭番地になります．

この場合，データの先頭はＤ100番地からなので，反対にして00Ｄ１と入れます．たとえば，先頭がＥ241番地だったら，41Ｅ２とするわけです．Ｚ80では番地は逆に入れることに注意してください．

PCGAIDを使って作ったパターンは，DBF０番地から入っています．この場所でじゃまにならないときはいいのですが，どこか他の番地へ移したいときにはこのパターン・データをブロック転送します．これは**リスト２**のようにします．データを破壊しない番地，たとえばＥ800あたりから書き込んでやります．

❶でＨＬレジスタにパターン・データ先頭番地を入れ，
❷でＤＥレジスタに新しく移動したデータの先頭番地を入れます．だから，この□内の数値は各自のデータ先頭に合わせて変えます．
❸はＢＣレジスタにデータ量を入れます．１つのパターンは８バイトで，128キャラクタで1,024バイトと，先頭に16バイトのキャラクタ作成のフラグがあるので合計1,040(410Ｈ）バイトです．
❹はＺ80特有のブロック転送命令です．
❺で終了したときBASICのコマンド待ちに戻ります．もし，モニタの状態に戻りたいときは，C3665Cとしてください．

新しいデータの番地がパターン・データのすぐそばで，前のデータを破壊してしまうときは，最終番地を指定してLDDR命令を使います．そして，モニタで**GE800 RET**とやればデータはブロック転送されます．このリスト２のプログラムは終わったらNOP（00）でも入れておいた方がいいでしょう．

普通，暴走してもマシン語が壊れることは少ないのですが，このプログラムが残っていると暴走時に実行して内容がみんな変わってしまうことがあります．

これでパターン・データとその転送プログラムが揃った

写真1　得点とキーの説明（漢字が書けるのはＰＣＧの強み）

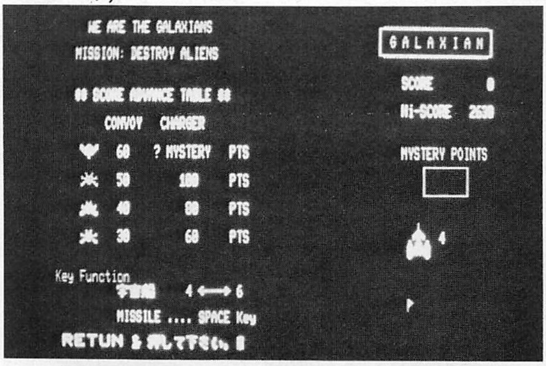


写真3　時折くる大編隊．本ものそっくり

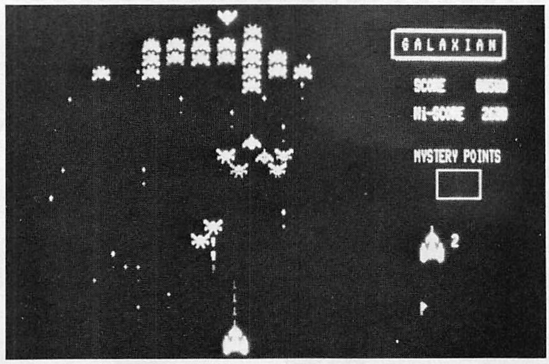


写真2　ゲーム・スタート直後．星も降ってきます．

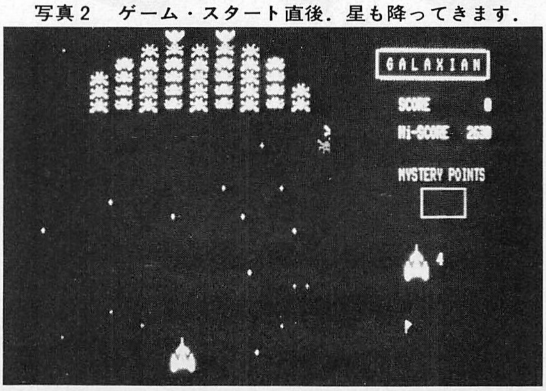


写真4　もう少しで1面が終わりそう！



ので，転送プログラムをコールすればＰＣＧにパターンのデータが入ります．

ただし，注意して欲しいのは，このデータ転送プログラムはキャラ・ジェネからのコピーのフラグは調べないので，PCGAIDにある**C**コマンドのようなことはできません．

データの内容がすべてＰＣＧに入ってしまいますから，キャラ・ジェネのあるコードを使いたいときにはその形を自分で作ってデータとして入れておく必要があります．

GALAXIAN 2 パターンの作り方

パターンをすべて作ってしまい，データをマシン語内に組み込んだ後でも一部をちょっと変更したいことがよくあるものです．この場合，1の方法でPCGAIDで変更して再びブロック転送をしてもいいのですが，これではあまりにも面倒なので，次に一部のパターンの変更の仕方について述べます．

まず，変更するパターンの番地を調べます．最初の16バイトはフラグ群ですから，一番最初のパターン（キャラクタ・コード80H）は，ギャラクシアンではD110番地から8バイト分ということで，キャラクタ・コード81HはD118番地から8バイトです．以下，1つコードが増えるごとに8バイトずつ増やしていけばいいので簡単に計算できると思います．

番地は16進数で表わすので，番地の末尾は0か8になることに注意してください．16進数では9の次は10ではなくてAですから…．筆者もマシン語を始めた頃はよくこれを間違えて暴走させたものでした．変更するパターンの先頭番地がわかったら次にその変更をします．

例としてD118番地からのキャラクタ・コード81Hを示します．図1のようにこれはギャラクシアンの胴体です．そして，そのデータは上から1行ずつ1バイトに入っていて合計8バイトです．

1バイトのデータでビットが1の部分にドットが描かれます．だから，ビットを調べてそれを16進数に直せばいいのですが，非常に面倒くさいことですし，わからない人も多いと思います．

私はこれが一目でわかるように**表1**のような対照表を使っています．この中から目的のパターンを探して上位，下位の順に数値を並べれば16進数が求められます．図1のパターンを見て**表1**で探していただければすぐ理解できると思います．

図1はずいぶん横に伸びていますが，そうです，これは40字の場合の表示で，PCGAIDのときはこの形で表示されていますが，使うときは80字ですから横は半分に圧縮されます．つまり，80字では横は縦の倍細かくなり，全部で640ドット分あるということになります．

GALAXIAN 3 音の出し方

ＰＣＧのもう一つの特長に音が出せるということがありますが，これがなかなかのくせもので，さっぱり思うような音が出てくれません．音の出し方には色々な方法があると思いますが，筆者の音の出し方の一例を**リスト3**に示しておきます．

これは音のデータ（ＳＮＤ１）の先頭番地をＨＬレジスタに入れて，SOUND１をコールすれば音が出ます．音のデータというのは，音の高さと音の出ている時間の２つが１組になっていて，０があると終了します．SNDONで音を出すためのスイッチを入れ，音階は４桁の16進数で最初に下位を次に上位をポート０ＣＨに入れます．

表1　ドット・パターンと16進数の対照表

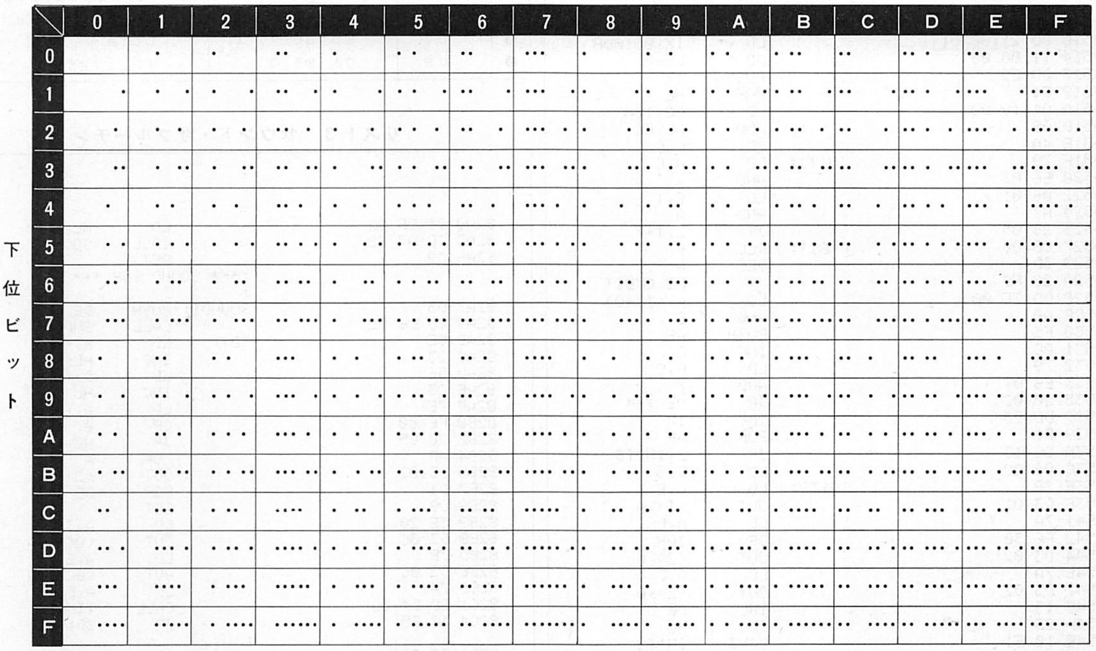
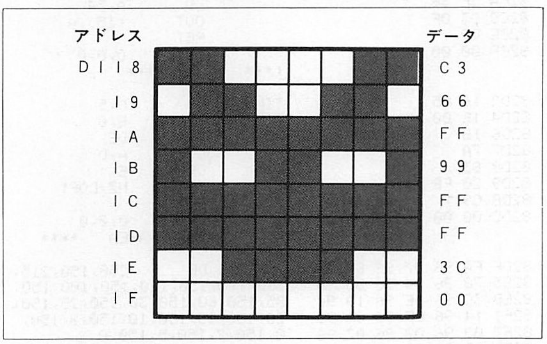


図1　パターンとデータの関係



下位の数値は変えてもあまり変化しないので，現在32にしてあり，音のデータは上位だけに入れています．音の出ている時間はDレジスタに数値を入れてTIMR 2 をコールしています．

この音ですが，なかなか思うような音が出てくれず，何より周波数の計算どおりの音が出てくれません．低音から高音へなめらかに上がっていくプログラムを作ったら途中にとんでもない音がいろいろ入ってきて，まるででたらめな曲を聴いているみたいになりました．

九十九電機で聴いたＰＣＧの音はかなりまともだったので，おかしいのは私のＰＣＧだけかもしれません．音は日によっても，スイッチを入れてからの時間によってもかなり変わってきます．２日間もつけっぱなしで酷使していると，反ストでも起こすのかノイズみたいな聴くに耐えない音になったりします．

ともかく，この音のプログラムで音が変なときは，1150行のＡレジスタに入れる数値（20H）をいろいろ変えてみてください．きれいな音の出る数値がみつかると思います．

第2章PCG版ギャラクシアン

ＰＣＧのすばらしさを理解してもらうには，やはり実際に何かやってみることだと思います．そこで今回は何にしようかと迷ったのですが，９月号に載せたギャラクシアンなら一度すでにプログラムしたものでもあるし簡単にできるだろうと（これが大間違いであったが……）プログラムしてみました．

GALAXIAN プログラムの入力

❶ BASIC

いつもどおりBASICの最後にはチェック・サムのプログラムが付属していて，ＲＵＮをすればまずチェック・サムの値が表示されます．

■チェック・サム

このチェック・サムというものがわからない方もいるみたいなので説明すると，これはマシン語の先頭番地から64個（バイト）ずつ合計を求めるもので，この数値が本の値と違っているときは，その64バイトの範囲のマシン語にどこかタイプ・ミスがあると思われるので，そこだけを調べれば済みます．これだと異常動作のあったときマシン語を端から端まで調べるなんて非能率的なことをしなくて済みます．

行番号9080のPRINTをLPRINTとすれば，プリンタに出力できます．チェック・サムがＯＫでしたら，行番号10と9000～9090を削除してください．

BASICプログラム中，カナ文字やグラフィック文字がでたらめに並んでいるところがありますが，このとおりに入力してください．ＰＣＧにデータが入れば意味のある表示をします．

リスト1　PCGデータ転送プログラム

```
                              ORG     0D510H
                        ;**PCG DATA TRANSMIT SUB**
                        ;
D510 DD 21 00 D1                LD      IX,0D100H
D514 11 00 00                   LD      DE,0
D517 DD E5                      PUSH    IX
D519 E1                         POP     HL
D51A 01 10 00                   LD      BC,10H
D51D 09                         ADD     HL,BC
D51E 4A                         LD      C,D
D51F 79                 INIT1:  LD      A,C
D520 E6 07                      AND     7
D522 06 01                      LD      B,1
D524 A7                         AND     A
D525 28 05                      JR      Z,$+7
D527 CB 00              CKBIT:  RLC     B
D529 3D                         DEC     A
D52A 20 FB                      JR      NZ,CKBIT
D52C DD 7E 00                   LD      A,(IX+0)
D52F A0                         AND     B
D530 F5                         PUSH    AF
D531 0C                         INC     C
D532 79                         LD      A,C
D533 E6 07                      AND     7
D535 20 02                      JR      NZ,$+4
D537 DD 23                      INC     IX
D539 F1                         POP     AF
D53A 28 13                      JR      Z,INIT3
D53C 06 08                      LD      B,8
D53E 7B                 INIT2:  LD      A,E
D53F D3 01                      OUT     (1),A
D541 7A                         LD      A,D
D542 F6 30                      OR      30H
D544 D3 02                      OUT     (2),A
D546 7A                         LD      A,D
D547 D3 02                      OUT     (2),A
D549 13                         INC     DE
D54A 23                         INC     HL
D54B 10 F1                      DJNZ    INIT2
D54D 18 14                      JR      INIT5
D54F 06 08              INIT3:  LD      B,8
D551 7E                 INIT4:  LD      A,(HL)
D552 D3 00                      OUT     (0),A
D554 7B                         LD      A,E
D555 D3 01                      OUT     (1),A
D557 7A                         LD      A,D
D558 F6 10                      OR      10H
D55A D3 02                      OUT     (2),A
D55C 7A                         LD      A,D
D55D D3 02                      OUT     (2),A
D55F 13                         INC     DE
D560 23                         INC     HL
D561 10 EE                      DJNZ    INIT4
D563 79                 INIT5:  LD      A,C
D564 87                         ADD     A,A
D565 30 B8                      JR      NC,INIT1
D567 C9                         RET

                                END

*SYMBOLS

D527 CKBIT    D51F INIT1    D53E INIT2    D54F INIT3
D551 INIT4    D563 INIT5
```

リスト2　ブロック転送プログラム

❶	E800	21 F0 DB	LD HL, 0DBF0H
❷	03	11 00 D1	LD DE, 0D100H
❸	06	01 10 04	LD BC, 0410H
❹	09	ED B0	LDIR
❺	0B	C3 E5 17	JP 17E5H

リスト3　サウンド・サブルーチン

```
                         ;
                         ;
                         ;
82A0 21 DF 82                    LD      HL,SND1
82A3 CD A7 82                    CALL    SOUND1
82A6 C9                          RET
                         ;*** SOUND SUB ***
                         ;
82A7 D5                  SOUND1: PUSH    DE
82A8 CD C6 82                    CALL    SNDON
82AB 5E                  SD1:    LD      E,(HL)
82AC 23                          INC     HL
82AD 56                          LD      D,(HL)
82AE 23                          INC     HL
82AF 7E                          LD      A,(HL)
82B0 FE 00                       CP      0
82B2 20 05                       JR      NZ,$+7
82B4 AF                          XOR     A
82B5 D3 02                       OUT     (2),A
82B7 D1                          POP     DE
82B8 C9                          RET
82B9 3E 20                       LD      A,32
82BB D3 0C                       OUT     (0CH),A
82BD 7B                          LD      A,E
82BE D3 0C                       OUT     (0CH),A
82C0 7A                          LD      A,D
82C1 CD D4 82                    CALL    TIMR2
82C4 18 E5                       JR      SD1
                         ;
82C6 3E 08               SNDON:  LD      A,8
82C8 D3 02                       OUT     (2),A
82CA 3E 36                       LD      A,54
82CC D3 0F                       OUT     (15),A
82CE C9                          RET
82CF 00 00 00                    DB      0,0,0
                         ;***  TIMER  ***
                         ;
82D2 16 05               TIMR1:  LD      D,5
82D4 1E 00               TIMR2:  LD      E,0
82D6 1B                  LOP1:   DEC     DE
82D7 7A                          LD      A,D
82D8 B3                          OR      E
82D9 20 FB                       JR      NZ,LOP1
82DB C9                          RET
82DC 00 00 00                    DB      0,0,0
                         ;****   DATA  AREA   ****
                         ;
82DF FA 96 D7 96 B9 96   SND1:   DB      250,150,215,
82E5 78 96 64 96 55 96   150,185,150,120,150,100,150,
82EB 3C 96 1E 96 19 96   85,150,60,150,30,150,25,150,
82F1 14 96 0F 96 0A 96   20,150,15,150,10,150,9,150,
82F7 09 96 08 96 07 96   8,150,7,150,5,150,0
82FD 05 96 00                                       END
*SYMBOLS
82D6 LOP1     82AB SD1     82DF SND1     82C6 SNDON
82A7 SOUND1   82D2 TIMR1   82D4 TIMR2
```

行番号500の［，］はカナシフト・モード，行番号670の［ー］はカナ・モードによるものです．

❷ マシン語の入力方法

マシン語の入力は誰にでもできるものと思っていたので，過去のゲームには何も書かなかったのですが，マシン語は初めてで入力の仕方のわからないという方が，かなりI/O編集部へ電話をしてきたとのことで，もうわかっていると思いますが簡単に書いておきます．モニタの説明はPCのユーザーズ・マニュアルp.64から書かれていますが，ちょっと説明不足の感じで初めての人にはわかりにくいと思います．

❶BASICで，**mon** **[RET]** とやると＊が表示されて，モニタ・モードになります．

❷ゲームの入力は最初から16進数のデータを入力していけばいいのですから，まず，**SD100** **[RET]** とやります．特定の番地（△△△△）から始めたければ**S** △△△△ **[RET]** とやればいいのです．左端に，**D100**と表示され，その右にその番地の内容が表示されます．ハイフンに続いて値を△△（16進数）とキーインすれば1つ右へずれ，次々と入れて8つ入れると改行されます（40字では4つで改行されます）．本のとおりに順番に入れていってください．

❸メモリの内容を変更せずに次へ移れないときは **[スペース]** キーを押します．

❹もし間違って入力したときは **[DEL]** キーを押せば1つ前に戻れます．

❺終了するときは **[RET]** を押します．

❻16進コード以外の数値を入れると，？を表示してコマンド・モードに戻ってしまうので，このときは**S** **[RET]** とやることで元の状態に戻れます．

❼すべて入力し終わって次に必要なのはテープへの書き込みです．BASICのCSAVEに相当するもので，**WD100**，**E5DF**とキーインしてテープレコーダを録音状態にしてから **[RET]** を押します．

❽テープにちゃんと入ったかのベリファイは，BASIC ではCLOAD？に相当するもので，LV [RET] とやります．?が出なければノーエラーです．

❾テープに入れたデータをＰＣに読み込むのはBASICのCLOADに相当するもので，L [RET] をやってテープをスタートさせます．

❿BASICへ戻るには，[CTRL] キーを押しながら [B] を押します．

初心者のうちはこれだけ知っていれば充分でしょう．マシン語に慣れてくればマニュアルを読んでもすぐにわかるようになると思います．

それにしてもこのＰＣのモニタはミニ・モニタともいうべきもので，後せめてレジスタ内容の表示と，ブレーク・ポイントの設定くらいの機能は欲しかった！

GALAXIAN 2 遊び方

９月号とまったく同様で，テンキーの [4] と [6] で左右に宇宙船（ビーム砲は宇宙船に改名されました）が動き，[スペース] キーでミサイル（ビームはミサイルに改名されました）が発射されます．ギャラクシアンの攻撃を避けながら全部撃破してください．

25面消し終わるとGIVE UPになります．5,000点を越えると宇宙船は１台追加されます．もし，[STOP] キーを押してしまったとき，それまでのハイスコアを保持しておきたかったら，ＲＵＮをせずに，GOTO 190 としてください．

GALAXIAN 3 PCG版の改良点・特徴

❶ギャラクシアンの形は本物そっくり．

❷背景の星が流れます．立体感をつけるために星の速度を３種類にしたのですが効果のほどは….

❸ギャラクシアンのミサイルは１機につき２発と減らし，画面上は最大でも８発までとして易しくなった(かな？)．

❹前のプログラムでは最後の１機が残っていても次の局面へ行ってしまいますが，今回は判定のルーチンをちゃんと付けたのですべて撃破して次の局面に進みます．

❺局面数は本物のように旗が増えていくようにしました．

❻ギャラクシアンの攻撃のとき音が出るようにしましたが，先にも述べたように各々のＰＣＧによってかなり音は違うと思うので，後で述べるように変更してみてください．筆者の場合，最初のうちはいい音ですが，酷使をしていると音が変わってきてしまいます．母船と護衛の降下時には音は出ません．

❼ギャラクシアンの形を見るため，またもやゲーム・センター通いをしてしまいました（ギャラクシアンは人気機種なのでまだ残っていた）が，コンボイの動きは１列ずつ逆になっていることを発見しそのようにしました．

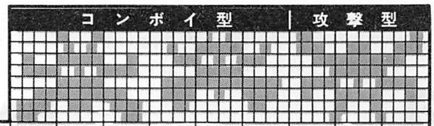
❽カラー，白黒の選択ができます．

❾ギャラクシアンの動きは前より大きくなりました．

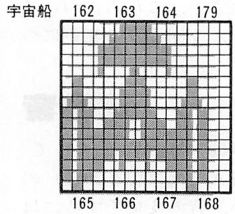
❿マシン語ルーチンの状態でも [ストップ] キーが効くようにしました．

宇宙船が描かれるまでの待ち時間は，イライラするという人もいるのですが，私としてはこのくらいの時間に少し首でも回したりして休憩した方がいいと思うので変更はしていません．

図２　パターンとキャラクタ・コードの関係



1～3行	128	129	130	131	132	133	134	135	136
4 行	137	138	139	140	141	142	143	144	145
5 行	146	147	148	149	150	151	152	153	154
母 船	155	156	157				202	203	204



ギャラクシアン爆発マーク

253	254	255

ギャラクシアン ミサイル	160
宇宙船のミサイル	161
星	158
旗	159

宇宙船爆発マーク（左から）

	上の行				下の行			
I	241	242	243	244	237	238	239	240
II	249	250	251	252	245	246	247	248



GALAXIAN 4 パターンの変更

ＰＣＧを使ったゲームの最大の特長は，パターンの変更が各自の思いどおりにできるということではないでしょうか．内容は同じゲームでもパターンを変えるとガラッと違ったゲームのように感じると思います．

たとえば，ギャラクシアンの形を変えてゴキブリアンなんてのを作ってはいかがですか．図２にパターンとキャラクタ・コードの関係を示します．この数値は10進数です．ここにないものに関してはBASIC のリストを見てもらえばわかると思います．

これで変更したいキャラクタ・コードを決めたら，第１章で述べた方法で変えることができます．もっとも，すべて変更する場合はこれでは大変なのでPCGAIDを使ってデータ・テープを作った方がいいでしょう．

このときはBASIC行番号180（ギャラクシアンのデータ転送プログラムをコールしている）を削除します．こうすればデータ・テープで入れたパターンはそのまま保存されます．

GALAXIAN 5 プログラムの解説

❶ マシン語

フローチャートは９月号とほとんど同じですから省略します．内容は前よりかなりすっきりしてきていると思います．各サブルーチンの後にはNOP（00）を３つ入れてあります．ＰＣＧパターン・データ，転送プログラム，星の移動，発生プログラム，ギャラクシアンが０機の判定プログ

ラム，音出しのプログラムなどが追加されています．マシン語サブルーチンの内容を表2に，ワーク・エリアを表3に，データ・エリアを表4に示します．

2 BASIC

前回とほとんど同様なのでわかると思います．

GALAXIAN 6 変更点

いつものように変更点を表5に示します．番号の頭に＊のついたものは2の倍数にしてください．これらについて詳しいことは9月号を見てください．今回は音の変更について詳しく書きます．

1 爆発音

爆発音の音出しサブルーチンはD607番地から始まっています．この音が変な音の場合D61A番地（現在20H）の値をいろいろと変えてみてください．音の変更をするにはデータ・エリアをいじります．表4を見てもらえばわかると思いますが，爆発音はSNDBM 1～3とSNDGBです．

このデータは音のデータと音の出ている時間の2つが1組になっています．変更した場合には最後に必ず0を入れておいてください．音のデータは数値の大きいほど低音，小さいほど高音が出ます．

2 飛行音

飛行音のサブルーチンはE154番地から始まっています．もし，このサブルーチンをいじるときは中にタイマも含まれていますから注意してください．タイマが外れると動きが速すぎて手がつけられなくなります．飛行音のおかしいときはE17E番地（現在20H）を変更してみてください．

飛行音のデータはE545番地から始まっている17個です．この個数は変えない方がいいでしょう．データは音の高さを規定しています．音の出る時間はタイマによって規定されるので変更は困難です．

GALAXIAN 7 アセンブラについて

今回，初めてアセンブラというものを使ってみました．日本マイコン学院から出ているROMタイプのものなので，テープ・ベースのものと違い，暴走させたときにも壊れないので助かります．このアセンブラを使った例はリスト1やリスト3を見てください．

1 アセンブラの利点

①何といっても後から付け加えるのが自由自在で，ハンド・アセンブルのときは改良のためいたるところにNOPを入れておいたのですが，今回のは見てわかるようにスッキリしているでしょう．

②ニモニックで入れるので，マシン語の表をいちいち見なくて済みます．もっともかなり覚えてしまっているので最初はニモニックをわざわざ打ち込むのは，わずらわしく感じました．

③ラベルが使えるので今までのようにワーク・エリアの表を片手になんてことはしなくてよくなりました．

④文法上のエラーが出る．ただし，プログラムのエラーは無理ですが．

⑤相対ジャンプの計算をしなくてよい．ハンド・アセンブルのときはこの計算ミスによる暴走が非常に多かった．

⑥ハード・コピーがすぐに取れる．

2 欠点

①ソース・テキストとオブジェクトがあるのでメモリはかなり食います．今回のギャラクシアン程度でほぼ1杯になります．

②この位の長いプログラムになると時間がずいぶんかかります．私の測ったところでは，テープにロードするのに6分18秒，アセンブルに12分55秒，プリント出力をしながらのアセンブルには何と1時間もかかりました．

特にバグ取りのときはたった1箇所の変更をして様子をみることが多いのですが，そのたびに13分近くもかかっているといい加減イヤになってきます．せめてディスク対応にしてもらえれば少しは助かるのですが…….

③BASICのような気軽さでプログラムできることは利点でもありますが，欠点でもあります．今回はこの気軽さでフローチャートを書かずにどんどんプログラムしてしまったら，虫さんの1個大隊が誕生してしまったのです．

④BASICから**mon**でアセンブラが起動してしまい，モニタへ行くにはさらに**mon**が必要です．帰りはもっと複雑で非常にめんどくさい．またテープによってはテープリードエラーが出やすくなります．ROMを抜いたり差したりばかりやっているのでそろそろピンがダメになるのではと心配です．

何はともあれ，マシン語をやる人にはアセンブラはやはり必需品だと思います．使い慣れてくるともう手離せません．このアセンブラはエディタも同居しているので大変使いやすいと思います．もうちょっと安いといいのですが……．

GALAXIAN 8 おわりに

9月号のギャラクシアンの焼き直しのつもりだったので，3日で完成の予定だったのですが，プログラムにかなり手を入れたことやアセンブラに不慣れなことも重なり，虫の大群が住みついてしまい，虫取りに10日近くかかってしまいました．

教訓――フローチャートはしっかり書こう!! まだ虫がすべていなくなったかどうかは自信がありません．ギャラクシアンが少なくなってくると，時としてギャラクシアンが画面の領域をはみ出して右側の枠まで侵入することがありますが，いまのところまた戻ってくるのでカンベンしてください．もし，アトリビュート・・エリアまで侵入して画面がでたらめになるようだったら早めにSTOPした方がいいと思います．暴走の恐れもあります．もしこうなったらゴメンナサイ．

筆者のHi-SCOREは15,400点でした．

■参考文献

1) Z80マイコンプログラミング・テクニック，電波新聞社
2) 芸夢狂人：〝ギャラクシアン〟，I/O，'80年9月号
3) ゲーム：ギャラクシアン

表 2 マシン語サブルーチン

ラベル	アドレス	内　容
DATRS	D510～D56F	PCGのデータ転送
MAIN	D570～D606	メイン・ルーチン
SOUND1	D607～D631	音出しのルーチン
TIMR1	D632～D63E	タイマ
	D63F～D761	画面に各種キャラクタを描くサブルーチン群
INITV	D762～D7AA	初期画面
RND	D7AB～D7CE	乱数を作るルーチン
STAINI	D7CF～D7FB	星を画面に描く初期
ADRS	D7FC～D81D	EレジスタにXの値，AレジスタにYの値を入れると，そこのアドレスを計算してHLに入れる．
OUT10	D81E～D84D	10進出力サブルーチン
MACL1	D84E～D870	クロック類の更新．スペースキーを戻したかの判定
GALDA	D871～D8D6	ギャラクシアンの踊り
GALMV	D8D7～D953	ギャラクシアンの左右移動
STASPD	D954～D972	星の移動速度を決める
STRMV	D973～D9C2	星の移動
STPMV	D9C3～DA05	宇宙船の左右移動
MISHT	DA06～DA2C	ミサイル発射
MISMV	DA2D～DB77	ミサイルの上昇，命中
MISRET	DB78～DB88	ミサイルを宇宙船の先端に付ける
COVB1	DB89～DBC8	コンボイ型のワークをクリアする
FLYB1	DBC9～DBF7	攻撃型のワークをクリアする
BOMPO	DBF8～DC1B	ギャラクシアンの爆発および，得点計算，表示
SPLS	DC1C～DC5E	宇宙船を追加するか
COLR	DC5F～DCD8	画面のカラー化
FLGA	DCD9～DCF8	攻撃するギャラクシアン数を決める．
FLTGAL	DCF9～DD9A	攻撃するギャラクシアンを決めてデータを移す．
EXG1	DD9B～DDC6	データのエクスチェンジ
GDOMA	DDC7～DDED	ギャラクシアン飛行のメイン・ルーチン
GADWN	DDEE～DEBE	〃　〃 のサブルーチン
GMSBTH	DEBF～DEE2	ギャラクシアンのミサイル発生
POTR1	DEE3～DF94	飛行のデータのポインタを変更する．
SPBOMB	DF95～E015	宇宙船の爆発
COVRET	E016～E05C	コンボイ型へ戻る．
SPINIT	E05D～E077	宇宙船爆発後の初期設定
GAZRO	E078～E09F	ギャラクシアン＝0のチェック
GAINIT	E0A0～E0BD	〃のときの初期設定
TVCLR	E0BE～E0D6	画面の1部のクリア
GMSDWN	E0D7～E153	ギャラクシアンのミサイル降下
TIMRFL	E154～E18C	ギャラクシアン飛行音
MSFLD1	E18D～E1E9	母船攻撃型のデータ移動
PRFLD1	E1EA～E242	護衛　〃　〃
PROLY	E243～E256	護衛のみの飛行データ移動
MSPBTH	E257～E2A6	母船と護衛の飛行するものを決める．
MSDMA	E2A7～E2D3	〃　〃 の飛行メイン・ルーチン
MSDWN	E2D4～E397	〃　〃　〃　サブルーチン
MYPOCL	E398～E3B1	ミステリーポイントを消す．
INIT1	E567～E58D	第1初期設定
INIT2	E58E～E5DE	第2　〃



表 3 ワーク・エリア

ラベル	アドレス	内　容
SPADR	E5DF, E5E0	宇宙船のアドレス
RNDDA	E5E1, E5E2	乱数値
STRWK	E5E3～E627	星のアドレス（23個分）
MACLK	E628	メイン・クロック他
CLK8	E629	8進クロック値
KEYPOP	E62A	スペースキーを戻したかのフラグ
GADIRC	E62B	ギャラクシアン移動方向値
STK1	E62C	〃　〃 の一時スタック
GLADR	E62D～E6F8	コンボイ型ギャラクシアンで，生死，アドレス，Y値，X値，タイプの順で，41機分．
MISAP	E723	ミサイル発射しているか
MISADR	E724, E725	ミサイル・アドレス
TICOT	E726	宇宙船表示までの待期カウンタ
DEHPR	E727	護衛を何台やられたか？
FLMWK	E728～E731	母船攻撃型で，生死，アドレス，Y値，X値，タイプ，コンボイ中の位置，飛行コース・ポインタ・アドレスの順．
FLGAWK	E732～E759	攻撃型ギャラクシアンで，上と同様，4機分．
FLPRWK	E75A～E76D	護衛攻撃型で，上と同様，2機分
GALCT	E76E	ギャラクシアン数
SCORE	E76F, E770	スコア
SPPLS	E771	宇宙船を追加したかのフラグ
SPCOT	E772	宇宙船の数
FLGACT	E773	飛行するギャラクシアンの数
BASRET	E774	BASICリターンのフラグ
DEHSP	E775	宇宙船をやられたかのフラグ
SCENE	E776	局面数
SCFLG	E777, E778	局面数を示す旗のアドレス
FLSDFG	E779	飛行音を出すかのフラグ
FLSDPT	E77A, E77B	飛行音のデータのポインタ
MSCRS1	E77C, E77D	上部母船飛行コース・データのポインタ
MSCRS2	E77E, E77F	下部母船飛行コース・データのポインタ
MPOAP	E780	ミステリーポイント表示しているかのフラグ
MPOCT	E781	〃　〃　表示カウンタ
GMSADR	E782～E79F	ギャラクシアン・ミサイルの生死とアドレス．8個分．

表 4 データ・エリア

ラベル	アドレス	内　容
GADAT	E3B2～E4AA	コンボイ型ギャラクシアンの初期設定データ
SNDBM1	E4AB～E4B7	宇宙船爆発音1
〃 2	E4B8～E4C2	〃　2
〃 3	E4C3～E4CF	〃　3
SNDGB	E4D0～E4DA	ギャラクシアン爆発音
	E4DB～E544	ギャラクシアン飛行コースのデータ群
SNDFLY	E545～E566	飛行音のデータ

表 5 変更点

変更する項目	番　地	現在の値
(1)全体のスピード	D633	05
*(2)宇宙船移動速度	D9C7	03
*(3)ミサイル速度	DA37	01
(4)ギャラクシアン飛行速度	D857	08
*(5)　〃　・ミサイル速度	E0DB	03
(6)宇宙船の数	E772	04

BASIC プログラム・リスト

```
10 GOTO9000
100 '╔═══════════════════════════════════════════════╗
110 'ワ   PCG GALAXIAN        VER 1.2              ワ
120 'ワ                              1980年11月3日   ワ
130 'ロ═══════════════════════════════════════════════╝
140 CLEAR100,&HD0FF
150 CONSOLE0,25,0,1:WIDTH80,25:COLOR7,0,0:PRINTCHR$(12)
160 LOCATE5,12:PRINT"YOUR TV = COLOR ? (Y or N)";
170 G$=INKEY$:IF G$="y"THEN C1=1:C2=2:C3=3:C4=4:C5=5:C6=6:C7=7:POKE&HDC9F,0 ELSE
 IF G$="n"THEN C1=7:C2=7:C3=7:C4=7:C5=7:C6=7:C7=7:POKE&HDC9F,&HC9 ELSE GOTO 170
180 DEFUSR2=&HD510:AA=USR2(0)
190 GOSUB410:GOSUB540
200 COLOR7,0,0:PRINTCHR$(12)
210 SC=0:GOSUB410
220 LINE(2,0)-(2,24)," ":LINE(50,0)-(53,24)," ",BF
230 DEFUSR3=&HDC67:AA=USR3(0)
240 DEFUSR1=&HD570:AA=USR1(0)
250 SC=PEEK(&HE770)*256+PEEK(&HE76F)
260 IF SC>HS THEN HS=SC:COLORC7:LOCATE71,7:PRINTUSING"#####";HS
270 F=PEEK(&HE774):IF F=1 THEN 280 ELSE GOTO 330
280 COLORC4:LOCATE15,10:PRINT"ｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒ"
290 LOCATE15,11:PRINT"**   G A M E   O V E R   **"
300 LOCATE15,12:PRINT"ﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝ"
310 FOR I=1TO3000:NEXT
320 GOTO 190
330 COLORC6:LOCATE15,10:PRINT"ｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒ"
340 LOCATE15,11:PRINT"***   G I V E    U P    ***"
350 LOCATE15,12:PRINT"ﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝ"
360 COLORC4:LOCATE18,15:PRINT"Y O U   W I N   !!!!"
370 COLORC5:LOCATE18,20:PRINT"P L A Y   A G A I N ?"
380 LOCATE16,22:COLORC7:PRINT"ｳｴｵﾔﾕﾖｯｰｳｴｱｲ ｵｶ ｷｸｹｺｻｼｽｾﾝﾀﾁﾂﾃ ";
390 IF INPUT$(1)<>CHR$(13)THEN390 ELSE GOTO 190
400 END
410 PRINTCHR$(12):COLORC6:LOCATE57,1:PRINT"ﾟ ｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒｒ口"
420 LOCATE57,2:PRINT"ワ G A L A X I A N ワ"
430 LOCATE57,3:PRINT"ロﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾝﾞ"
440 COLORC7:LOCATE61,5:PRINTUSING"SCORE       #####";SC
450 LOCATE61,7:PRINTUSING"Hi-SCORE  #####";HS
460 COLORC3:LOCATE61,10:PRINT"MYSTERY POINTS"
470 COLORC4:LOCATE64,11:PRINT"ﾟ═══════╪":LOCATE64,12:PRINT"ワ        ワ":LOCATE64,13:
PRINT"ロ══════ﾞ"
480 COLORC6:LOCATE65,12:PRINT"       "
490 COLORC2:LOCATE63,15:PRINT"｡"
500 COLORC5:LOCATE62,16:PRINT"｢｣､ｳ":LOCATE62,17:PRINT"･ｦｧｨ"
510 COLORC7:LOCATE67,16:PRINT"4"
520 COLORC2:LINE(62,21)-(76,22)," ",BF:LOCATE62,21:PRINT"ｨ"
530 RETURN
540 COLORC3:LOCATE11,1:PRINT"WE ARE THE GALAXIANS"
550 LOCATE9,3:PRINT"MISSION: DESTROY ALIENS"
560 COLORC7:LOCATE9,6:PRINT"** SCORE ADVANCE TABLE **"
570 COLORC5:LOCATE14,8:PRINT"CONVOY    CHARGER"
580 LOCATE16,10:PRINT"60     ? MYSTERY   PTS"
590 LOCATE16,12:PRINT"50       100     PTS"
600 LOCATE16,14:PRINT"40        80     PTS"
610 LOCATE16,16:PRINT"30        60     PTS"
620 COLORC6:LOCATE10,10:PRINT"ｲ ﾍ":COLORC2:LOCATE10,12:PRINT"ｷ ﾄ"
630 COLORC3:LOCATE10,14:PRINT"|▌":COLORC1:LOCATE10,16:PRINT"__▬"
640 COLORC4:LOCATE6,19:PRINT"Key Function"
650 COLORC5:LOCATE16,20:PRINT"ﾄﾅﾆﾇﾈﾉ    4 ◢═══ 6"
660 LOCATE16,22:PRINT"MISSILE .... SPACE Key"
670 LOCATE7,24:COLORC7:PRINT"ｳｴｵﾔﾕﾖｯｰｳｴｱｲ ｵｶ ｷｸｹｺｻｼｽｾﾝﾀﾁﾂﾃ ";
680 IF INPUT$(1)<>CHR$(13)THEN680 ELSE RETURN
9000 ' ***   CHECK SUM   ***
9010 CLEAR100,&HD0FF:WIDTH40,25:CONSOLE0,25,0,0:PRINTCHR$(12)
9020 PRINT"PCG GALAXIAN  CHECK SUM"
9030 PRINT"START=D100":S=&HD100:PRINT"END=  E5DF":E=&HE5DF:T=64:PRINT
9040 X=0:R=S+T:IF R>=E THEN P=E+1
9050 FOR I=S TO R-1:Y=PEEK(I)
9060 X=X+Y:NEXT
9070 S$=HEX$(S):S=S+T:E$=HEX$(R-1):X$=STRING$(4-LEN(HEX$(X)),"0")+HEX$(X)
9080 PRINT S$;" - ";E$;" =";X$
9090 IF S>=E THEN END ELSE GOTO 9040
```

チェック・サム

D100 - D13F = 1146	D540 - D57F = 1A80	D980 - D9BF = 19E6	DDC0 - DDFF = 262E	E200 - E23F = 1A8F
D140 - D17F = 161D	D580 - D5BF = 24CF	D9C0 - D9FF = 2487	DE00 - DE3F = 1BD3	E240 - E27F = 219A
D180 - D1BF = 16FB	D5C0 - D5FF = 2100	DA00 - DA3F = 1BC9	DE40 - DE7F = 254B	E280 - E2BF = 26AD
D1C0 - D1FF = 1AD7	D600 - D63F = 17FF	DA40 - DA7F = 1CC8	DE80 - DEBF = 1A07	E2C0 - E2FF = 1B6A
D200 - D23F = 16ED	D640 - D67F = 17BE	DA80 - DABF = 1C55	DEC0 - DEFF = 1D04	E300 - E33F = 1CB5
D240 - D27F = 17D0	D680 - D6BF = 1815	DAC0 - DAFF = 1DBA	DF00 - DF3F = 1C44	E340 - E37F = 23A9
D280 - D2BF = 10A2	D6C0 - D6FF = 1EC5	DB00 - DB3F = 1E9A	DF40 - DF7F = 1CD6	E380 - E3BF = 1B44
D2C0 - D2FF = 1179	D700 - D73F = 1D52	DB40 - DB7F = 1E5B	DF80 - DFBF = 2256	E3C0 - E3FF = 11A6
D300 - D33F = 0C48	D740 - D77F = 22D9	DB80 - DBBF = 179F	DFC0 - DFFF = 1FF3	E400 - E43F = 0F43
D340 - D37F = 1A89	D780 - D7BF = 1CCC	DBC0 - DBFF = 1BB1	E000 - E03F = 1C0A	E440 - E47F = 11B1
D380 - D3BF = 2820	D7C0 - D7FF = 1EFC	DC00 - DC3F = 2027	E040 - E07F = 1992	E480 - E4BF = 0FA3
D3C0 - D3FF = 1BA0	D800 - D83F = 167C	DC40 - DC7F = 193A	E080 - E0BF = 1966	E4C0 - E4FF = 0D75
D400 - D43F = 08E7	D840 - D87F = 197B	DC80 - DCBF = 1A06	E0C0 - E0FF = 17D6	E500 - E53F = 1AF6
D440 - D47F = 11D3	D880 - D8BF = 1C96	DCC0 - DCFF = 1935	E100 - E13F = 1DA8	E540 - E57F = 0B52
D480 - D4BF = 095C	D8C0 - D8FF = 196E	DD00 - DD3F = 1B22	E140 - E17F = 1AD2	E580 - E5BF = 1DE2
D4C0 - D4FF = 0A0C	D900 - D93F = 1433	DD40 - DD7F = 215A	E180 - E1BF = 1BC6	E5C0 - E5DF = 09D8
D500 - D53F = 14C8	D940 - D97F = 1835	DD80 - DDBF = 1543	E1C0 - E1FF = 1CB5	

マシン語プログラム・リスト

```
D100 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D110 : 00 00 30 3F 00 0F 7E 78 C3 66 FF 99 FF FF 3C 00
D120 : 00 00 0C FC 00 F0 7E 1E 00 30 1F 60 3F 1F 00 00
D130 : 66 FF 99 FF FF 3C 00 00 00 0C F8 06 FC F8 00 00
D140 : 78 7E 0F 00 3F 30 00 00 00 3C FF FF 99 FF 66 C3
D150 : 1E 7E F0 00 FC 0C 00 00 00 00 10 1F 00 0F 7F 78
D160 : 42 24 FF 99 FF FF 18 00 00 00 08 F8 00 F0 FE 1E
D170 : 00 18 0F 00 7F 3F 0E 00 24 FF 99 FF FF 18 00 00
D180 : 00 18 F0 00 FE FC 70 00 78 7F 0F 00 1F 10 00 00
D190 : 00 18 FF FF 99 FF 24 42 1E FE F0 00 F8 08 00 00
D1A0 : 01 00 30 3F 01 0F 7C 70 81 C3 FF 99 FF FF 3C 00
D1B0 : 80 00 0C FC 80 F0 3E 0E 00 60 1F 01 3F 7F 30 00
D1C0 : C3 FF 99 FF FF 3C 00 00 00 06 F8 00 FC FE 0C 00
D1D0 : 70 7C 0F 01 3F 30 00 01 00 3C FF FF 99 FF C3 81
D1E0 : 0E 3E F0 80 FC 0C 00 80 60 78 7E 1F 07 01 00 00
D1F0 : 3C FF FF FF 99 FF 18 18 06 1E 7E F8 E0 80 00 00
D200 : 00 00 10 38 10 00 00 00 E0 FC FF FC F0 C0 C0 C0
D210 : 00 00 3C 3C 3C 18 18 00 00 00 00 06 06 06 06 0F
D220 : 00 00 01 01 01 30 30 78 1F 7F FF FF 06 0F 1F 1F
D230 : 80 E0 F8 F8 08 00 80 81 DC DC DF DF DF DF FC 30
D240 : 3F F9 F9 FF DF 10 00 00 C3 F3 FF FF BF 8F 03 00
D250 : B0 B0 B0 B0 B0 B0 F0 C0 3F 30 30 3F 30 30 30 00
D260 : F0 0C 0C F0 C0 30 0C 00 3F 30 30 3F 30 30 3F 00
D270 : FC 00 00 C0 00 00 FC 00 3F 01 01 01 01 01 01 00
D280 : FC 80 80 80 80 80 80 00 30 30 30 30 30 30 0F 00
D290 : 0C 0C 0C 0C 0C 0C F0 00 30 3C 33 30 30 30 30 00
D2A0 : 0C 0C 0C CC 3C 0C 0C 00 00 00 00 00 00 C0 C0 E0
D2B0 : FF 81 81 81 81 81 81 FF 01 0F 03 07 00 03 0C 0F
D2C0 : 80 E0 00 C0 64 F8 00 F0 33 33 FB 33 33 3C F0 30
D2D0 : FF 33 FF 33 FF 30 30 30 30 18 18 18 18 18 0C 07
D2E0 : 00 00 00 00 00 08 30 E0 00 3F 00 00 01 03 01 00
D2F0 : 00 FC 30 C0 80 00 80 60 7F 03 03 03 03 03 03 03
D300 : FC 00 C0 30 00 00 00 00 03 3F 03 01 30 18 06 01
D310 : 00 F0 00 80 60 00 00 E0 00 0C 18 30 30 30 1A 0C
D320 : 00 00 00 18 0C 0C 08 00 00 00 00 00 00 70 50 70
D330 : 03 FF C0 0F 03 3F 03 07 00 FC 0C C0 00 F0 00 00
D340 : 03 FF C3 3F 33 3F 33 3F 00 FC 0C F0 30 F0 30 F0
D350 : 0C 7F 63 6B FF 63 6B C3 36 26 43 00 BF 33 33 3F
D360 : 00 00 00 07 1F 7F 70 00 18 18 7E FF 99 FF FF 3C
D370 : 00 00 00 E0 F8 FE 0E 00 FF 81 81 81 81 81 81 FF
D380 : FF 81 81 81 81 81 81 FF FF 81 81 81 81 81 81 FF
D390 : FF 81 81 81 81 81 81 FF FF 81 81 81 81 81 81 FF
D3A0 : FF 81 81 81 81 81 81 FF FF 81 81 81 81 81 81 FF
D3B0 : FF 81 81 81 81 81 81 FF FF 81 81 81 81 81 81 FF
D3C0 : FF 81 81 81 81 81 81 FF FF 81 81 81 81 81 81 FF
D3D0 : FF 81 81 81 81 81 81 FF FF 81 81 81 81 81 81 FF
D3E0 : FF 81 81 81 81 81 81 FF 18 18 18 1F 00 00 00 00
D3F0 : 18 18 18 18 18 18 18 18 00 00 00 FF 66 00 00 00
D400 : 18 18 18 F8 00 00 00 00 00 00 00 1F 18 18 18 18
D410 : 00 00 00 FF 00 00 00 00 00 00 66 FF 00 00 00 00
D420 : 00 00 00 F8 18 18 18 18 00 30 0C FF 0C 30 00 00
D430 : 00 0C 30 FF 30 0C 00 00 00 03 0C 30 3F 00 00 00
D440 : C0 C0 C0 C0 F8 C0 C0 00 0F 18 00 00 03 0C 1F 00
D450 : F8 0C 0C 38 C0 00 FC 00 1F 30 30 1F 30 30 1F 00
D460 : F8 0C 0C F8 0C 0C F8 00 0F 18 30 30 30 18 0F 00
D470 : F0 18 0C 0C 0C 18 F0 00 C0 E0 00 0C 03 0C 00 00
D480 : 00 30 CC 03 00 00 2C E0 00 0F 03 0F 00 32 1E 00
D490 : 00 00 80 00 04 0C 03 00 00 00 03 02 00 0C 3E 10
D4A0 : 00 C0 C0 03 00 00 03 03 00 3C 0C 03 C0 03 00 C0
D4B0 : 00 00 00 00 00 30 C0 30 0C 00 00 30 CC 30 27 18
D4C0 : 08 10 20 C0 00 00 00 03 20 18 04 03 00 30 43 00
D4D0 : 00 01 0C 00 CC 23 1C 03 C3 0F 30 0C 33 00 00 30
D4E0 : 0C 04 00 00 00 C0 30 00 C3 00 30 00 03 04 08 00
D4F0 : F3 30 03 CF 00 00 0E 00 00 30 0C 03 80 D3 0C 30
D500 : 20 0C 00 33 C8 0C 10 21 03 31 CC 00 30 0C CA 07
D510 : DD 21 00 D1 11 00 00 DD E5 E1 01 10 00 09 4A 79
D520 : E6 07 06 01 A7 28 05 CB 00 3D 20 FB DD 7E 00 A0
D530 : F5 0C 79 E6 07 20 02 DD 23 F1 28 13 06 08 7B D3
D540 : 01 7A F6 30 D3 02 7A D3 02 13 23 10 F1 18 14 06
D550 : 08 7E D3 00 7B D3 01 7A F6 10 D3 02 7A D3 02 13
D560 : 23 10 EE 79 87 30 B8 C9 00 00 00 00 00 00 00 00
D570 : CD 67 E5 CD 8E E5 CD 62 D7 CD 4E D8 CD 54 E1 CD
D580 : 71 D8 CD D7 D8 CD 73 D9 3A 26 E7 3D FE 02 20 09
D590 : 2A DF E5 CD B5 D6 CD 7C DB FE 00 28 05 32 26 E7
D5A0 : 18 16 CD C3 D9 CD 06 DA CD 2D DA 3A 28 E6 E6 1F
D5B0 : 20 03 CD F9 DC CD 57 E2 CD 98 E3 CD D7 E0 CD A7
D5C0 : E2 CD C7 DD DB 09 FE FE C8 3A 74 E7 FE 01 20 04
D5D0 : AF D3 02 C9 3A 75 E7 FE 01 20 05 CD 5D E0 18 99
D5E0 : CD 78 E0 3A 6E E7 FE 00 20 8F AF D3 02 3A 76 E7
D5F0 : 3C 32 76 E7 FE 1A 20 06 3E 05 32 74 E7 C9 CD A0
D600 : E0 C3 76 D5 00 00 00 D5 CD 26 D6 5E 23 56 23 7E
D610 : FE 00 20 05 AF D3 02 D1 C9 3E 20 D3 0C 7B D3 0C
D620 : 7A CD 34 D6 18 E5 3E 08 D3 02 3E 36 D3 0F C9 00
D630 : 00 00 16 05 1E 00 1B 7A B3 20 FB C9 00 00 00 36
D640 : 80 23 36 81 23 36 82 C9 36 83 23 36 84 23 36 85
D650 : C9 36 86 23 36 87 23 36 88 C9 36 89 23 36 8A 23
D660 : 36 8B C9 36 8C 23 36 8D 23 36 8E C9 36 8F 23 36
D670 : 90 23 36 91 C9 36 92 23 36 93 23 36 94 C9 36 95
D680 : 23 36 96 23 36 97 C9 36 98 23 36 99 23 36 9A C9
D690 : 36 9B 23 36 9C 23 36 9D C9 36 CA 23 36 CB 23 36
D6A0 : CC C9 36 00 23 36 00 23 36 00 23 C9 36 FD 23 36
D6B0 : FE 23 36 FF C9 36 A5 23 36 A6 23 36 A7 23 36 A8
D6C0 : D5 11 88 FF 19 D1 36 B3 2B 36 A4 2B 36 A3 2B 36
D6D0 : A2 C9 36 ED 23 36 EE 23 36 EF 23 36 F0 D5 11 88
D6E0 : FF 19 D1 36 F4 2B 36 F3 2B 36 F2 2B 36 F1 C9 36
D6F0 : F5 23 36 F6 23 36 F7 23 36 F8 D5 11 88 FF 19 D1
D700 : 36 FC 2B 36 FB 2B 36 FA 2B 36 F9 C9 CD A2 D6 36
```

```
D710 : 00 D5 11 85 FF 19 CD A2 D6 36 00 19 23 36 00 D1
D720 : C9 21 E2 F8 36 E7 23 36 E8 23 18 14 21 E2 F8 36
D730 : E5 23 36 E6 23 18 09 21 E2 F8 36 E9 23 36 EA 23
D740 : 36 EB 23 36 EC 23 36 EB 23 36 EC 3E 01 32 80 E7
D750 : 3E 80 32 81 E7 C9 21 E2 F8 CD A2 D6 C3 A2 D6 00
D760 : 00 00 CD CF D7 21 62 F5 CD 9E D7 21 EA F4 CD 9E
D770 : D7 21 72 F4 CD 9E D7 21 FE F3 06 07 CD 5A D6 23
D780 : 23 10 F9 21 8A F3 06 05 CD 75 D6 23 23 10 F9 21
D790 : 16 F3 CD 90 D6 21 1E F3 CD 90 D6 C3 56 D7 06 09
D7A0 : CD 3F D6 23 23 10 F9 C9 00 00 00 E5 D5 2A E1 E5
D7B0 : 54 5D 19 19 7D 84 67 11 81 00 19 22 E1 E5 7C D1
D7C0 : E1 C9 CD AB D7 FE 30 D8 DE 30 18 F9 00 00 00 06
D7D0 : 17 0E 01 DD 21 E3 E5 CD C2 D7 C6 04 5F 79 CD FC
D7E0 : D7 DD 75 00 DD 23 DD 74 00 DD 23 CD 54 D9 DD 77
D7F0 : 00 DD 23 36 9E 0C 10 DF C9 00 00 00 C5 D5 21 00
D800 : 00 11 78 00 06 08 CB 0F 30 01 19 EB 29 EB 05 20
D810 : F5 11 00 F3 19 D1 16 00 19 C1 C9 00 00 00 01 10
D820 : 27 CD 3A D8 01 E8 03 CD 3A D8 01 64 00 CD 3A D8
D830 : 0E 0A CD 3A D8 7D C6 30 12 C9 C5 37 3F 3E 30 ED
D840 : 42 38 03 3C 18 F9 C1 09 12 13 C9 00 00 00 21 28
D850 : E6 34 3A 29 E6 3C FE 08 20 01 AF 32 29 E6 3A 2A
D860 : E6 FE 01 C0 DB 09 FE FF C0 AF 32 2A E6 C9 00 00
D870 : 00 3A 28 E6 E6 0F C0 DD 21 2D E6 06 27 DD 7E 00
D880 : FE 01 28 08 11 06 00 DD 19 10 F2 C9 DD 6E 01 DD
D890 : 66 02 E5 CD A2 D6 E1 DD 7E 05 CB 67 20 1B FE 01
D8A0 : 20 05 CD 48 D6 18 0C FE 02 20 05 CD 63 D6 18 03
D8B0 : CD 7E D6 DD CB 05 E6 18 CB FE 11 20 05 CD 3F D6
D8C0 : 18 0C FE 12 20 05 CD 5A D6 18 03 CD 75 D6 DD CB
D8D0 : 05 A6 18 B0 00 00 00 3A 28 E6 E6 1F C0 DD 21 2D
D8E0 : E6 06 29 DD 6E 01 DD 66 02 3A 2B E6 FE 00 20 2B
D8F0 : 23 DD 75 01 DD 74 02 DD 34 04 DD 7E 00 FE 01 20
D900 : 42 DD 7E 04 FE 30 20 05 3E 01 32 2C E6 0E 03 23
D910 : 23 2B 7E 23 77 2B 0D 20 F8 18 26 2B DD 75 01 DD
D920 : 74 02 DD 35 04 DD 7E 00 FE 01 20 17 DD 7E 04 FE
D930 : 04 20 04 AF 32 2C E6 0E 03 23 7E 2B 77 23 0D 20
D940 : F8 36 00 11 06 00 DD 19 10 99 3A 2C E6 32 2B E6
D950 : C9 00 00 00 CD AB D7 FE 03 38 04 DE 03 18 F8 FE
D960 : 01 20 03 3E 3F C9 FE 02 20 03 3E 1F C9 3E 0F C9
D970 : 00 00 00 3A 28 E6 E6 07 C0 DD 21 E3 E5 06 17 DD
D980 : 4E 02 3A 28 E6 A1 28 08 11 03 00 DD 19 10 F0 C9
D990 : DD 6E 00 DD 66 01 7E FE 9E 20 02 36 00 11 78 00
D9A0 : 19 7C FE FE 20 0B CD C2 D7 C6 04 5F 3E 01 CD FC
D9B0 : D7 DD 75 00 DD 74 01 7E FE 00 20 02 36 9E 18 C8
D9C0 : 00 00 00 3A 28 E6 E6 03 C0 DB 00 FE EF 28 25 FE
D9D0 : BF C0 2A DF E5 7D FE 70 C8 E5 CD 0C D7 E1 23 22
D9E0 : DF E5 E5 CD B5 D6 E1 11 11 FF 3A 23 E7 FE 00 C0
D9F0 : 19 36 A1 C9 2A DF E5 7D FE 43 C8 E5 CD 0C D7 E1
DA00 : 2B 18 DC 00 00 00 3A 2A E6 FE 00 C0 3A 23 E7 FE
DA10 : 00 C0 DB 09 FE BF C0 3E 01 32 2A E6 32 23 E7 2A
DA20 : DF E5 11 11 FF 19 22 24 E7 C9 00 00 00 3A 23 E7
DA30 : FE 01 C0 3A 28 E6 E6 01 C0 2A 24 E7 7E FE A1 20
DA40 : 02 36 00 11 88 FF 19 7C FE F2 20 03 CD 78 DB 22
DA50 : 24 E7 7E FE 80 28 16 FE 83 28 12 FE 81 28 0D FE
DA60 : 84 28 09 FE 82 28 04 FE 85 20 0B 2B 2B CD 89 DB
DA70 : 01 1E 00 C3 F8 DB FE 89 28 16 FE 8C 28 12 FE 8A
DA80 : 28 0D FE 8D 28 09 FE 8B 28 04 FE 8E 20 0B 2B 2B
DA90 : CD 89 DB 01 28 00 C3 F8 DB FE 92 28 16 FE 95 28
DAA0 : 12 FE 93 28 0D FE 96 28 09 FE 94 28 04 FE 97 20
DAB0 : 0B 2B 2B CD 91 DB 01 32 00 C3 F8 DB FE 86 28 0A
DAC0 : FE 87 28 05 FE 88 20 0B 2B 2B CD C9 DB 01 3C 00
DAD0 : C3 F8 DB FE 8F 28 0A FE 90 28 05 FE 91 20 0B 2B
DAE0 : 2B CD C9 DB 01 50 00 C3 F8 DB FE 98 28 0A FE 99
DAF0 : 28 05 FE 9A 20 12 2B 2B CD D1 DB 3A 27 E7 3C 32
DB00 : 27 E7 01 64 00 C3 F8 DB FE 9B 28 0A FE 9C 28 05
DB10 : FE 9D 20 0B 2B 2B CD 99 DB 01 3C 00 C3 F8 DB FE
DB20 : CA 28 0A FE CB 28 05 FE CC 20 40 2B 2B DD 21 28
DB30 : E7 CD BA DB 3A 27 E7 FE 00 20 0F 01 C8 00 E5 CD
DB40 : 21 D7 E1 AF 32 27 E7 C3 F8 DB FE 01 20 0A 01 90
DB50 : 01 E5 CD 2C D7 E1 18 EB FE 02 28 05 AF 32 27 E7
DB60 : C9 01 20 03 E5 CD 37 D7 E1 18 D8 FE 00 28 03 FE
DB70 : 9E C0 36 A1 C9 00 00 00 AF 32 23 E7 2A DF E5 11
DB80 : 11 FF 19 36 A1 C9 00 00 00 DD 21 2D E6 06 22 18
DB90 : 0E DD 21 F9 E6 06 05 18 06 DD 21 17 E7 06 02 DD
DBA0 : 7E 00 FE 01 28 08 11 06 00 DD 19 10 F2 C9 DD 7E
DBB0 : 01 BD 20 F2 DD 7E 02 BC 20 EC DD 36 00 00 3A 6E
DBC0 : E7 3D 32 6E E7 C9 00 00 00 06 04 DD 21 32 E7 18
DBD0 : 06 06 07 DD 21 28 E7 DD 7E 00 FE 01 28 08 11 0A
DBE0 : 00 DD 19 10 F2 C9 DD 7E 01 BD 20 F2 DD 7E 02 BC
DBF0 : 20 EC C3 BA DB 00 00 00 E5 CD AC D6 21 D0 E4 CD
DC00 : 07 D6 E1 CD A2 D6 2A 6F E7 09 22 6F E7 11 A0 F5
DC10 : CD 1E D8 CD 1C DC C3 78 DB 00 00 00 2A 6F E7 3A
DC20 : 71 E7 FE 00 C8 7C FE 13 20 07 A7 7D FE 88 D8 18
DC30 : 04 A7 FE 13 D8 3A 72 E7 3C 32 72 E7 C6 30 11 C4
DC40 : FA 12 AF 32 71 E7 06 0F 3E 20 D3 40 16 06 CD 34
DC50 : D6 AF D3 40 16 03 CD 34 D6 10 ED C9 00 00 00 11
DC60 : 78 00 77 19 10 FC C9 3E 08 06 19 21 53 F3 CD 5F
DC70 : DC 06 19 21 51 F3 CD 5F DC 06 19 21 57 F3 CD 5F
DC80 : DC 3E 04 06 19 21 54 F3 CD 5F DC 3E 33 06 19 21
DC90 : 56 F3 CD 5F DC 3E 36 06 19 21 58 F3 CD 5F DC 00
DCA0 : 00 3E C8 32 55 F3 3E 48 32 CD F3 3E 68 32 45 F4
DCB0 : 21 BD F4 3E 28 06 03 CD 5F DC 3E 88 06 06 CD 5F
DCC0 : DC 3E C8 06 06 CD 5F DC 3E 48 06 05 CD 5F DC 3E
DCD0 : A8 06 02 C3 5F DC 00 00 00 21 73 E7 3A 6E E7 FE
DCE0 : 1D 20 04 3E 02 77 C9 FE 0D 20 04 3E 03 77 C9 FE
DCF0 : 06 C0 3E 04 77 C9 00 00 00 CD D9 DC 4E DD 21 2D
DD00 : E6 FD 21 32 E7 06 22 DD 7E 00 FE 01 28 08 11 06
DD10 : 00 DD 19 10 F2 C9 FD 7E 00 FE 00 28 09 0D C8 11
```

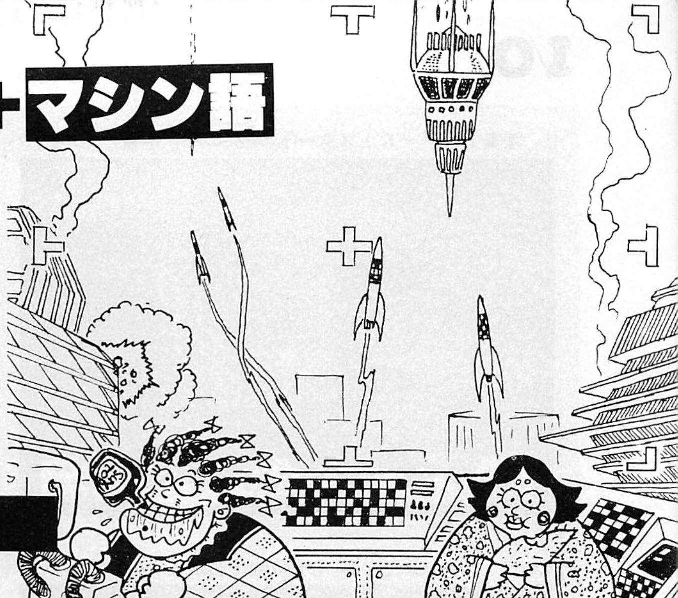
マシン語プログラム・リスト

```
DD20 : 0A 00 FD 19 18 F0 FD 36 00 01 DD 6E 01 DD 66 02
DD30 : E5 CD A2 D6 E1 A7 DD 7E 04 FE 1B 30 0D 11 87 FF
DD40 : DD 7E 04 3D E5 21 DB E4 18 0B 11 89 FF DD 7E 04
DD50 : 3C E5 21 E0 E4 FD 75 08 FD 74 09 E1 FD 77 04 19
DD60 : FD 75 01 FD 74 02 DD 7E 03 3D FD 77 03 DD 7E 05
DD70 : FD 77 05 CD 87 DD DD E5 E1 FD 75 06 FD 74 07 DD
DD80 : 36 00 00 0D C8 18 87 CB 47 20 03 C3 6C D6 CB 4F
DD90 : 20 03 C3 87 D6 C3 51 D6 00 00 00 21 3C E7 18 08
DDA0 : 21 46 E7 18 03 21 50 E7 11 32 E7 18 0B 21 5A E7
DDB0 : 18 03 21 64 E7 11 28 E7 06 0A 4E 1A 77 79 12 23
DDC0 : 13 10 F7 C9 00 00 00 3A 29 E6 FE 01 C0 CD EE DD
DDD0 : CD 9B DD CD EE DD CD 9B DD CD A0 DD CD EE DD CD
DDE0 : A0 DD CD A5 DD CD EE DD C3 A5 DD 00 00 00 DD 21
DDF0 : 32 E7 DD 7E 00 FE 01 C0 DD 6E 01 DD 66 02 CD A2
DE00 : D6 DD 7E 03 3C DD 77 03 FE 06 20 12 E5 21 45 E5
DE10 : 22 7A E7 3E 01 32 79 E7 E1 CD E3 DE 18 07 FE 0E
DE20 : 20 03 CD 35 DF DD 7E 04 DD 6E 08 DD 66 09 86 37
DE30 : 3F FE 34 38 07 DD 36 00 00 C3 16 E0 23 DD 75 08
DE40 : DD 74 09 DD 77 04 5F DD 7E 03 F5 CD FC D7 F1 FE
DE50 : 0F 28 04 FE 11 20 07 F5 E5 CD BF DE E1 F1 FE 17
DE60 : 20 0A E5 CD AA DE E1 FE 01 CA 95 DF DD 7E 03 FE
DE70 : 18 20 0A E5 CD 90 DE E1 FE 01 CA 95 DF DD 7E 03
DE80 : FE 19 28 B1 DD 75 01 DD 74 02 DD 7E 05 C3 87 DD
DE90 : 06 03 7E FE A5 28 0C FE A6 28 08 FE A7 28 04 FE
DEA0 : A8 20 03 3E 01 C9 23 10 E9 C9 06 03 7E FE A3 28
DEB0 : 04 FE A4 20 03 3E 01 C9 23 10 F1 C9 00 00 00 23
DEC0 : FD 21 82 E7 06 08 FD 7E 00 FE 00 28 08 11 03 00
DED0 : FD 19 10 F2 C9 FD 36 00 01 FD 75 01 FD 74 02 C9
DEE0 : 00 00 00 CD ED DE DD 75 08 DD 74 09 C9 DD 7E 04
DEF0 : FE 1B 38 1F CD 7E DF FE 01 20 04 21 F1 E4 C9 FE
DF00 : 02 20 04 21 21 E5 C9 FE 03 20 04 21 39 E5 C9 21
DF10 : 09 E5 C9 CD 7E DF FE 01 20 04 21 E5 E4 C9 FE 02
DF20 : 20 04 21 2D E5 C9 FE 03 20 04 21 15 E5 C9 21 FD
DF30 : E4 C9 00 00 00 CD 3F DF DD 75 08 DD 74 09 C9 CD
DF40 : 88 DF FE 01 20 04 21 E5 E4 C9 FE 02 20 04 21 FD
DF50 : E4 C9 FE 03 20 04 21 15 E5 C9 FE 04 20 04 21 2D
DF60 : E5 C9 FE 05 20 04 21 F1 E4 C9 FE 06 20 04 21 09
DF70 : E5 C9 FE 07 20 04 21 21 E5 C9 21 39 E5 C9 CD AB
DF80 : D7 FE 05 D8 DE 05 18 F9 CD AB D7 FE 08 D8 DE 08
DF90 : 18 F9 00 00 00 AF 32 32 E7 C5 2A DF E5 06 05 3E
DFA0 : C8 CD 06 E0 CD D2 D6 21 AB E4 CD 07 D6 2A DF E5
DFB0 : CD 0C D7 21 B8 E4 CD 07 D6 2A DF E5 3E 48 CD 06
DFC0 : E0 CD EF D6 21 C3 E4 CD 07 D6 2A DF E5 10 D0 16
DFD0 : 96 CD 34 D6 CD 0C D7 3A 9F DC FE C9 28 04 3E A8
DFE0 : 18 02 3E E8 CD 06 E0 C1 3A 72 E7 3D 32 72 E7 C6
DFF0 : 30 11 C4 FA 12 3A 72 E7 FE 00 3E 01 20 04 32 74
E000 : E7 C9 32 75 E7 C9 E5 C5 21 1D FE 06 02 CD 5F DC
E010 : C1 E1 C9 00 00 00 DD 2A 38 E7 DD 36 00 01 DD 6E
E020 : 01 DD 66 02 DD 7E 04 FE 31 38 12 FE D0 30 11 2B
E030 : 2B 2B 2B 3D 3D 3D 3D FE 31 30 F4 18 13 FE 04 D0
E040 : 23 23 23 23 3C 3C 3C 3C FE D0 30 F4 FE 04 38 F0
E050 : DD 75 01 DD 74 02 DD 77 04 C9 00 00 00 3E F0 32
E060 : 26 E7 AF 32 75 E7 32 23 E7 2A 24 E7 36 00 21 5A
E070 : FE 22 DF E5 C9 00 00 00 21 28 E7 06 07 7E FE 01
E080 : C8 11 0A 00 19 10 F6 21 2D E6 06 29 7E FE 01 C8
E090 : 11 06 00 19 10 F6 AF 32 6E E7 D3 02 C9 00 00 00
E0A0 : 2A 77 E7 36 9F 23 7D FE 26 20 02 2E 8F 22 77 E7
E0B0 : CD BE E0 CD 8E E5 16 FA C3 34 D6 00 00 00 21 89
E0C0 : F2 11 78 00 0E 19 19 E5 AF 06 36 77 23 10 FC E1
E0D0 : 0D 20 F3 C9 00 00 00 3A 28 E6 E6 03 C0 DD 21 82
E0E0 : E7 06 08 DD 7E 00 FE 01 28 08 11 03 00 DD 19 10
E0F0 : F2 C9 DD 6E 01 DD 66 02 7E FE A0 20 02 36 00 11
E100 : 78 00 19 7C FE FE 20 0B 7D FE B5 38 06 DD 36 00
E110 : 00 18 D7 DD 75 01 DD 74 02 C5 06 01 E5 CD AC DE
E120 : E1 C1 FE 01 28 0D E5 C5 06 01 CD 92 DE C1 E1 FE
E130 : 01 20 0D DD 36 00 00 CD 99 DF 21 6E E7 34 18 AA
E140 : 7E FE 00 20 04 36 A0 18 A1 FE 9E 20 9D 36 A0 18
E150 : 99 00 00 00 3A 29 E6 FE 01 20 07 3A 79 E7 FE 01
E160 : 28 03 C3 32 D6 2A 7A E7 CD 26 D6 7E 23 22 7A E7
E170 : 23 FE 00 20 07 AF D3 02 32 79 E7 C9 F5 3E 20 D3
```

```
E180 : 0C F1 D3 0C 16 05 CD 34 D6 C9 00 00 00 DD 21 17
E190 : E7 18 04 DD 21 1D E7 DD 36 00 00 FD 21 28 E7 FD
E1A0 : 36 05 05 FD 36 00 01 DD 6E 01 DD 66 02 FD 75 01
E1B0 : FD 74 02 DD 7E 03 FD 77 03 DD E5 E1 FD 75 06 FD
E1C0 : 74 07 DD 7E 04 FD 77 04 FE 1B 30 05 21 E0 E4 18
E1D0 : 03 21 DB E4 FD 75 08 FD 74 09 FD 75 3A FD 74 3B
E1E0 : FD 75 44 FD 74 45 C9 00 00 00 DD 21 F9 E6 18 04
E1F0 : DD 21 05 E7 0E 02 06 03 FD 21 5A E7 DD 7E 00 FE
E200 : 01 28 08 11 06 00 DD 19 10 F2 C9 FD 77 00 DD 36
E210 : 00 00 DD 7E 01 FD 77 01 DD 7E 02 FD 77 02 DD 7E
E220 : 03 FD 77 03 DD 7E 04 FD 77 04 FD 36 05 03 DD E5
E230 : E1 FD 75 06 FD 74 07 0D C8 11 0A 00 FD 19 18 C3
E240 : 00 00 00 DD 21 F9 E6 06 05 0E 01 CD F8 E1 FD 7E
E250 : 04 C3 C8 E1 00 00 00 3A 6E E7 FE 22 D0 3A 28 E7
E260 : FE 00 C0 3A 5A E7 FE 00 C0 3A 64 E7 FE 00 C0 CD
E270 : AB D7 FE 6F C0 AF 32 27 E7 DD 21 28 E7 CD ED DE
E280 : 22 7C E7 CD 3F DF 22 7E E7 3A 17 E7 FE 01 20 06
E290 : CD 8D E1 C3 EA E1 3A 1D E7 FE 01 C2 43 E2 CD 93
E2A0 : E1 C3 F0 E1 00 00 00 3A 29 E6 FE 03 C0 CD AD DD
E2B0 : CD D4 E2 CD AD DD CD B2 DD CD D4 E2 CD B2 DD 21
E2C0 : 7C E7 34 23 23 34 CD D4 E2 21 7C E7 35 23 23 35
E2D0 : C9 00 00 00 DD 21 28 E7 DD 7E 00 FE 01 C0 DD 6E
E2E0 : 01 DD 66 02 CD A2 D6 DD 7E 03 3C DD 77 03 FE 06
E2F0 : 20 05 2A 7C E7 18 07 FE 0E 20 09 2A 7E E7 DD 75
E300 : 08 DD 74 09 DD 7E 04 DD 6E 08 DD 66 09 86 FE 34
E310 : 38 24 DD 36 00 00 DD 2A 2E E7 CD 1A E0 3A 17 E7
E320 : FE 00 28 06 2A 18 E7 CD 90 D6 3A 1D E7 FE 00 C8
E330 : 2A 1E E7 C3 90 D6 23 DD 75 08 DD 74 09 DD 77 04
E340 : 5F DD 7E 03 F5 CD FC D7 F1 FE 0E 28 04 FE 11 20
E350 : 07 E5 F5 CD BF DE F1 E1 FE 17 20 10 E5 CD AA DE
E360 : E1 FE 01 20 07 DD 36 00 00 C3 99 DF DD 7E 03 FE
E370 : 18 20 0A E5 CD 90 DE E1 FE 01 CA 65 E3 DD 7E 03
E380 : FE 19 28 8E DD 75 01 DD 74 02 DD 7E 05 FE 03 CA
E390 : 87 D6 C3 99 D6 00 00 00 3A 80 E7 FE 00 C8 3A 81
E3A0 : E7 3D 32 81 E7 FE 00 C0 AF 32 80 E7 C3 56 D7 00
E3B0 : 00 00 01 92 F4 03 2A 01 01 72 F4 03 0A 01 01 16
E3C0 : F4 02 26 12 01 FE F3 02 0E 12 01 0A F5 04 2A 01
E3D0 : 01 EA F4 04 0A 01 01 82 F5 05 2A 01 01 62 F5 05
E3E0 : 0A 01 01 8E F4 03 26 11 01 76 F4 03 0E 11 01 06
E3F0 : F5 04 26 11 01 EE F4 04 0E 11 01 7E F5 05 26 11
E400 : 01 66 F5 05 0E 11 01 12 F4 02 22 02 01 02 F4 02
E410 : 12 02 01 8A F4 03 22 01 01 7A F4 03 12 01 01 02
E420 : F5 04 22 01 01 F2 F4 04 12 01 01 7A F5 05 22 01
E430 : 01 6A F5 05 12 01 01 0E F4 02 1E 12 01 06 F4 02
E440 : 16 12 01 0A F4 02 1A 02 01 86 F4 03 1E 11 01 82
E450 : F4 03 1A 01 01 7E F4 03 16 11 01 F6 F4 04 16 11
E460 : 01 FA F4 04 1A 01 01 FE F4 04 1E 11 01 76 F5 05
E470 : 1E 11 01 72 F5 05 1A 01 01 6E F5 05 16 11 01 8A
E480 : F3 01 12 03 01 8E F3 01 16 13 01 92 F3 01 1A 03
E490 : 01 96 F3 01 1E 13 01 9A F3 01 22 03 01 16 F3 00
E4A0 : 16 00 01 1E F3 00 1E 00 00 00 00 08 28 0F 19 64
E4B0 : 19 96 0F C8 14 00 00 00 FA 14 C8 14 B4 1E 64 14
E4C0 : 00 00 00 50 19 14 1E 28 14 0A 14 0F 0F 00 00 00
E4D0 : 1E 08 14 05 08 0A 04 0A 00 00 00 FE FF FF 00 00
E4E0 : 02 01 01 00 00 00 01 01 02 02 02 01 01 01 00 00
E4F0 : FF 00 FF FF FE FE FE FF FF FF 00 00 01 00 01 02
E500 : 02 02 03 02 01 00 00 FF FE 00 FF FE FE FE FD FE
E510 : FF 00 00 01 02 00 01 03 04 04 04 03 02 01 00 FF
E520 : FD 00 FF FD FC FC FC FD FE FF 00 01 03 00 00 01
E530 : 02 03 01 00 FF FE FD FF 00 00 00 FF FE FD FF 00
E540 : 01 02 03 01 00 09 0C 0F 10 11 12 13 14 15 16 17
E550 : 18 19 1A 1B 1C 1D 00 00 00 00 00 00 00 00 00 00
E560 : 00 00 00 00 00 00 00 AF 32 29 E6 32 2A E6 32 74
E570 : E7 32 80 E7 21 00 00 22 6F E7 3E 01 32 71 E7 32
E580 : 76 E7 3E 04 32 72 E7 21 18 FD 22 77 E7 C9 21 5A
E590 : FE 22 DF E5 21 B2 E3 11 2D E6 01 F6 00 ED B0 AF
E5A0 : 32 2B E6 32 2C E6 32 23 E7 32 75 E7 32 79 E7 3E
E5B0 : F0 32 26 E7 32 27 E7 DD 21 28 E7 06 07 DD 36 00
E5C0 : 00 11 0A 00 DD 19 10 F5 3E 29 32 6E E7 3E 01 32
E5D0 : 73 E7 21 82 E7 06 08 36 00 23 23 23 10 F9 C9 00
```

PC-8001 BASIC+マシン語

ミサイルコマンダー



東大PCユーザーズグループ　**安倍昌宏**



ーム・センターでお馴染みのTVゲームをPC-8001に載せたものが，すでに幾つか発表されていますが，これもその1つで，『ミサイル・コマンダー』のPC版といったところです．ミサイル・コマンダーは，インベーダーやヘッド・オン・タイプのゲームとは違って，カーソル(照準)を操作して発射目標を定めるという方法を使っているので，他のゲームとはまた違った面白さがあります．実際のゲーム・マシンでは画像が美しく，効果音なども迫力があり，とてもPCでは追いつきませんが，できるだけ本格的なものにしてみたので，ぜひ挑戦してください．

ゲームの説明

このゲームは大変有名（と筆者は信じます）なゲームですから，知っている人も多いと思いますが，本物と違う点もあるので少し説明します．

写真1を見てください．左端，中央，右端にある丘がミサイル基地で，手持ちのミサイルはすべてここに表示されており，左から順に11基，9基，10基と，全部で30基のミサイルがあります．そして，基地の間にある水色のところが都市で，ここを守るわけです．

まず，敵ミサイル（『ATTACK MISSILE』と呼ぶ）が飛跡を残しながら，地上の都市や基地目掛けて降下してくるので（ラインの先端にミサイルがあると思ってください），カーソル（照準）をその飛来先に移動し，発射キーを押して基地からミサイルを打ち上げます．発射基地はキーによって選択することができ，1度に10発まで発射可能です．

ミサイルが発射されると，カーソルのあった位置にクロスマークが表示され，ミサイルはこのマーク目指して敵ミサイルと同じように，青色の飛跡を残しながら進みます（**写真2**）．

ミサイルがマークに到達すると，そこで爆発して爆雲を生じ，このとき，敵ミサイルがうまくこの爆雲に包まれれば，敵ミサイルも爆発，消滅し，空からの攻撃を防ぐことができます(**写真3**)．

敵ミサイルはいろいろなところからいろいろな方向へ飛来し，さらに，分裂することもあるので，カーソルをすばやく縦横に動かして迎撃用ミサイルを次々に打ち上げないと，敵ミサイルは地上に達し，ミサイル基地や都市を破壊してしまいます．

ここでポイントになるのは，発射された迎撃ミサイルが爆雲を広げるまでの時間と，敵ミサイルがそこに到達する時間を見計らってカーソルをセットすることで，また場面が進むにつれて敵ミサイルの降下速度が増すので，それも計算に入れなければなりません．

破壊された都市は画面上から消え，ミサイル基地もミサイルがすべてなくなって；機能が停止してしまいます（**写真4**）．ミサイルが全部なくなると手も足も出なくなるので，都市を守ることだけでなく，基地を守ることにも気を配らなければなりません．

また，戦闘中に破壊衛星が出現し，ミサイルを落としたりしますが，これも爆雲に包まれれば爆発します(**写真5**)．破壊衛星の投下するミサイルは，低い位置で分裂するので防ぐのが難しく，衛星が現れたらすぐ消してしまうことです．もしそれが間に合わなくてミサイルが投下されたら，分裂して増える前に消した方がいいでしょう．

敵ミサイルがすべて消滅すると，その場面は終了で，次に得点の計算に入ります．このとき，残った都市が1つも

写真１　ゲーム・スタート（中央の十字がカーソル）

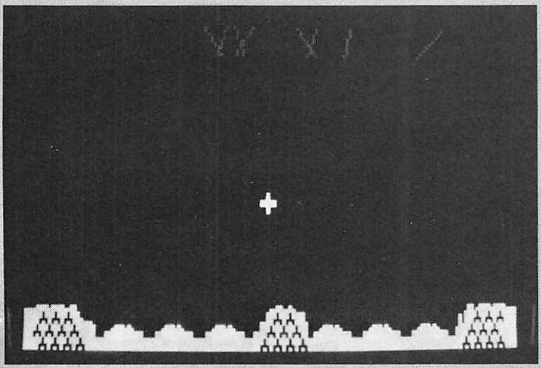


写真２　カーソルでセットした位置に迎撃ミサイルが飛んでいく

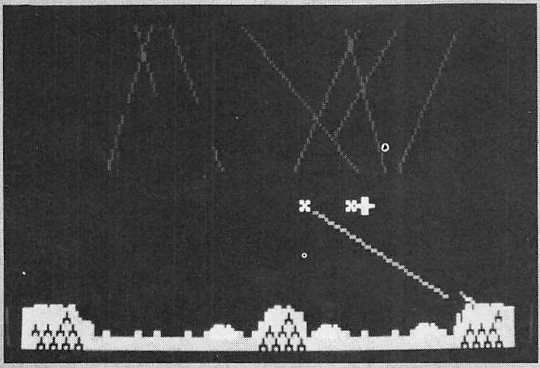


写真３　迎撃ミサイルによって爆雲発生

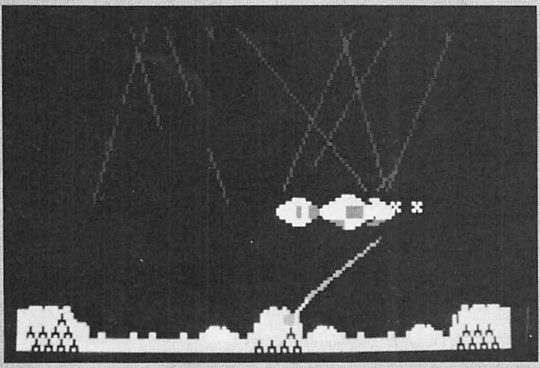


写真４　ミサイルが基地や都市へ次々と襲いかかる

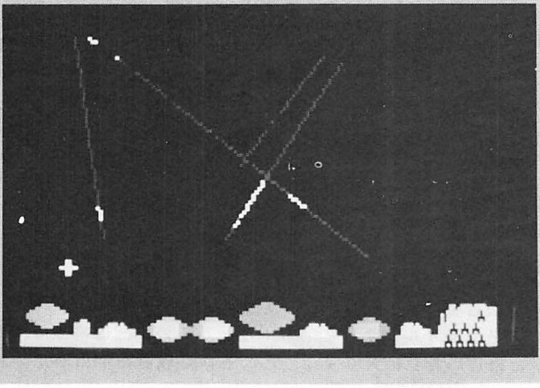


写真５　右中央にあるのが破壊衛星

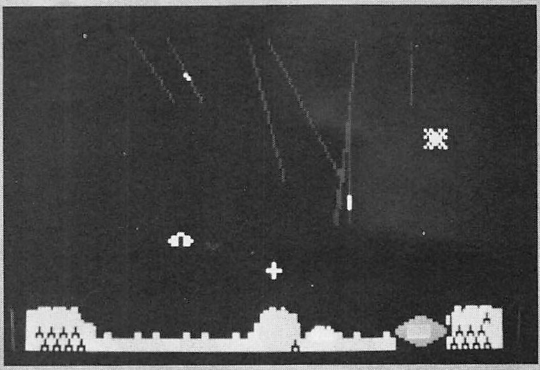


写真６　スコア，得点，ゲームの説明



写真７　早めに手を打つのが，このゲームのコツ

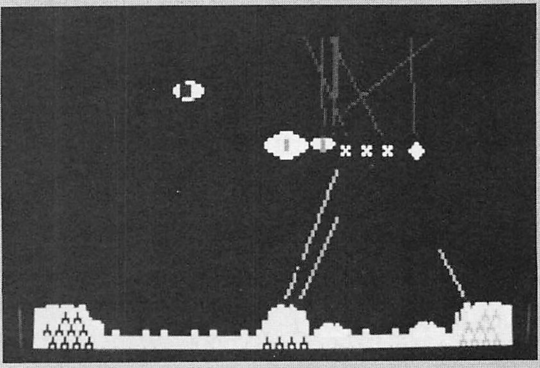
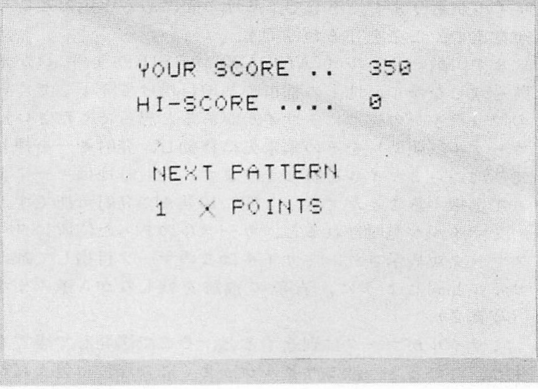


写真８　各面に移るごとに，得点が表示される



ないとゲーム・オーバーで，フィナーレを通して再ゲーム入力待ちとなります．

プログラムの実行

まず，プログラムをRUNさせると，**写真6**のようにタイトル，HI-SCORE，SCORE，説明文などが表示されます．

ここでＹキーを押すとゲームが開始され，画面が切り変わって，現在のSCORE，次の場面の得点の倍率などを表示した後，けたたましい効果音(?)とともに，都市，ミサイル基地，カーソルが現れます．

音が出ているときは，まだ敵ミサイルは発生しませんが，カーソルは動かすことができ，迎撃しやすい位置に移動させ待機します．

キー操作は，ゲーム・スタート入力時の説明文にも表示されますが，カーソルの移動は良く使われるテンキーの２(↓)，４(←)，６(→)，８(↑)で，ミサイルの発射が右側のシフト・キーの左３つのキーです．それぞれ，左端，中央，右端の基地に対応しています．

このキーにした理由は，３つ同じポートから読み出すことができるので，キーが押されている間，ミサイルが連続発射してしまうのを防ぐのに便利だったからです．また，カーソルは縦方向，横方向のキーを同時に押すことにより，斜めに進めることもできます．

遊び方は先ほどの『ゲームの説明』でだいたいわかると思いますが，ゲームをやっていて少し変に思われるかもしれない点がいくつかあるので，それをつけ加えておきます．

まず，

- ●都市が全部消滅してしまったのに，後から後から敵ミサイルが発生して来て終わらない．
- ●画面上の敵ミサイルを全部消しても場面が終わらない．

ということですが，これは１場面に発生する敵ミサイルの数が決まっていて，それが０にならないうちは終了できないことによります．

つまり，場面が進むにつれて，後から発生するミサイルの数が多くなるのですが，そのとき，早い内にやられてしまうと，敵ミサイルが全部出てしまうまで待っていなければならないことになります．さらに，発生する間隔も一度にたくさん現われないように少し広くとってあるので，しばらく発生してこないことも起こるわけです．

それから，画面上に爆雲が多くなるとカーソルの移動速度が遅くなります．私のプログラムのせいかもしれませんが，爆雲を明滅させるのはさすがのマシン語でも難儀のようです．

得点

得点は敵ミサイルをやっつけると25点，破壊衛星は100点で，これに場面が進むにつれて倍率が掛けられます．初めは１倍ですが，２面消すごとに２倍，３倍と増えていき，最高６倍にまでなります．６倍の場面は１回でこれを終えると最初に戻り，再び１倍から始めます．

結局，11面終えて１回りということですが，場面が進んでいくと敵の攻撃が強くなるので，１回りするのは至難の技だと思います．また，倍率が変わるのに応じて敵ミサイルの色や基地の色も変化します．

ところで得点の他に，このゲームでは１面ごとにボーナス点が与えられます．残った手持ちのミサイル１つに付き５点，防衛され残った都市１つに付き100点で，やはりこれにも倍率が掛けられます．

また，スコアが１万点増すごとに都市が１つ与えられます（ただし，６つあるときはそれ以上増えません）．

表１　BASICプログラムの説明

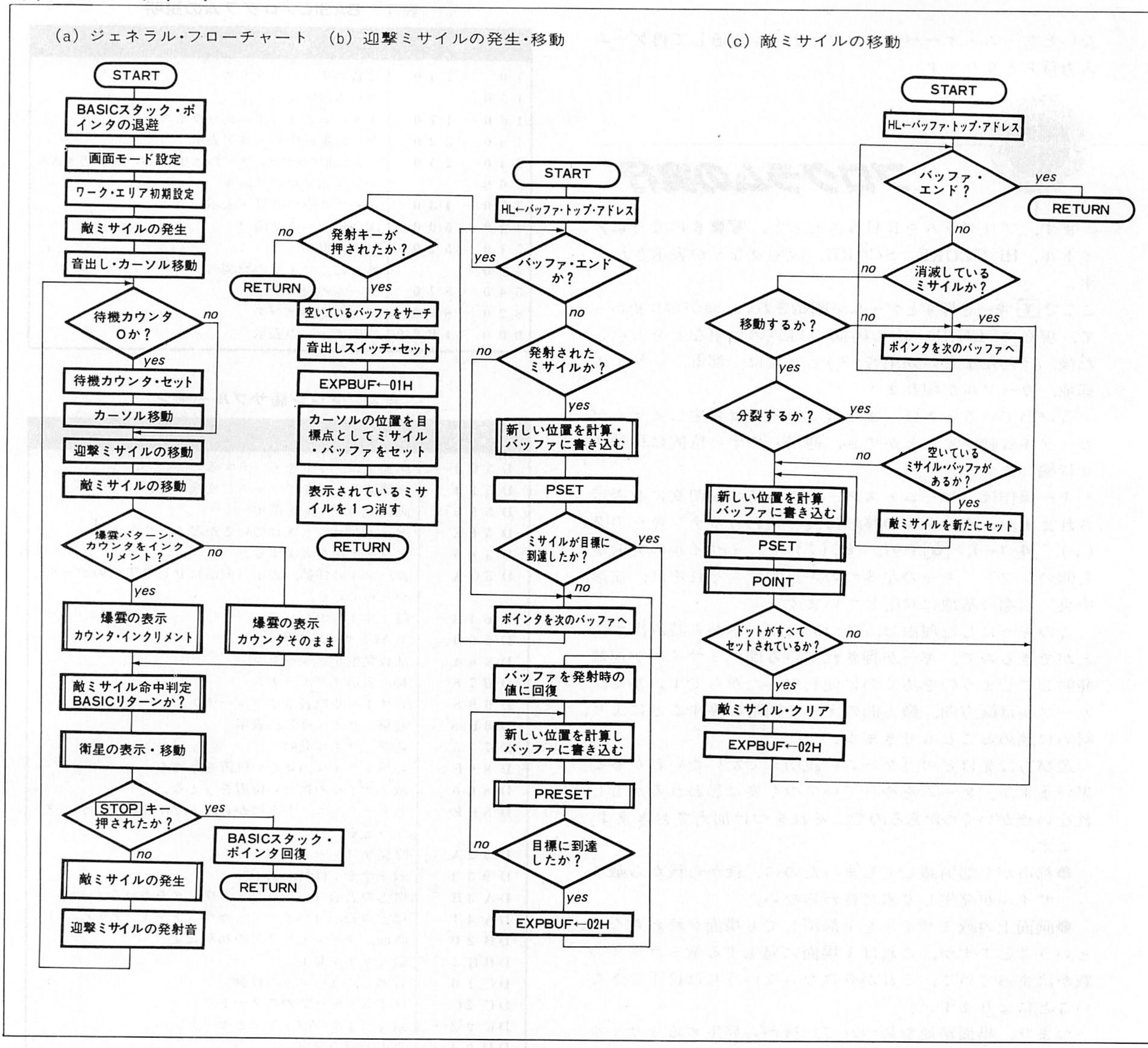
行番号	処理
10　～140	変数のイニシャライズ
150	ゲーム説明文へ
160～170	１ゲームごとのイニシャライズ
180～230	ゲーム進行中のスコア表示
240～250	マシン語プログラムワーク・エリアへのデータ書き込み
260	マシン語プログラムへ
270～430	ボーナス点の計算・表示
440～500	次の場面への準備
510～520	都市の追加
530	残ったミサイルの数調べ
540～810	ゲームの説明表示
820～890	場面設定用データ
900～1060	フィナーレの表示

表２　マシン語サブルーチン

スタート・アドレス	説明
D500～	画面上にドットをセットする．
D524～	座標値X，Yをアドレスに変換する．
D555～	画面上のドットを消す．
D56E～	ドットがセットされているかどうかを与える．
D585～	ドットで絵を表示する．
D5CA～	カーソルの移動，表示（前部にPLOT用のデータが入っている）
D61B～	敵ミサイルのクリア
D660～	WAITサブルーチン
D66A～	乱数発生サブルーチン
D678～	都市表示サブルーチン
D698～	ミサイル基地表示サブルーチン
D6F5～	迎撃ミサイル移動，表示
D775～	迎撃ミサイル発射
D80E～	迎撃ミサイルの新しい位置を与える．
D866～	敵ミサイルの新しい位置を与える．
D8BE～	スタート，エンド座標から，ミサイル移動のデータを与える．
D92A	除算サブルーチン
D933～	敵ミサイル移動，表示
DA3B～	爆雲の表示（パターン・カウンタをあげない）
DA41～	爆雲の表示（パターン・カウンタをインクリメント）
DB20～	画面，ワーク・エリアの初期設定
DBB1	敵ミサイル発生
DC10	音出し，カーソルの移動
DC2C～	メイン・ループのスタート
DC92～	敵ミサイルのバッファをセット
DD04～	都市消滅の判定
DD66	ミサイル基地消滅の判定
DE18～	衛星の表示，移動
DEFB	ミサイルの投下
DF92～	単音を出すサブルーチン
DFA7～	爆発の音出し



図1　フローチャート



プログラムについて

タイトルの表示，得点の計算表示など画面40字モードのところはすべてBASICが行なっており，その内容を**表1**に示します．乱数の初期値は毎回同じにならないように，内蔵クロックの秒を取り出して使用しています．

マシン語の部分はアセンブラを使って作ったので案外小さくまとまり，それでも約2.8Kくらいの大きさです．

図1はマシン語プログラムのフローチャートですが，詳しく書くと大変な量になるので，主なものの概略だけにしてあります．サブルーチンの中でPLOTというのは，BCレジスタに中心位置X，Yを入れ，HLペア・レジスタにプロットする絵のデータの格納先頭番地を入れてCALLすると，その位置を中心にドット単位で絵を描くサブルーチンで，カーソルの表示に使っています．

図1の中のEXPサブルーチンは爆雲を表示するもので，EXPBUFはそのバッファのことです．パターン・カウンタの意味は01Hでクロス・マークの表示，02H～0DHとインクリメントされるにつれて対応する大きさの爆雲が表示されるようになっています．

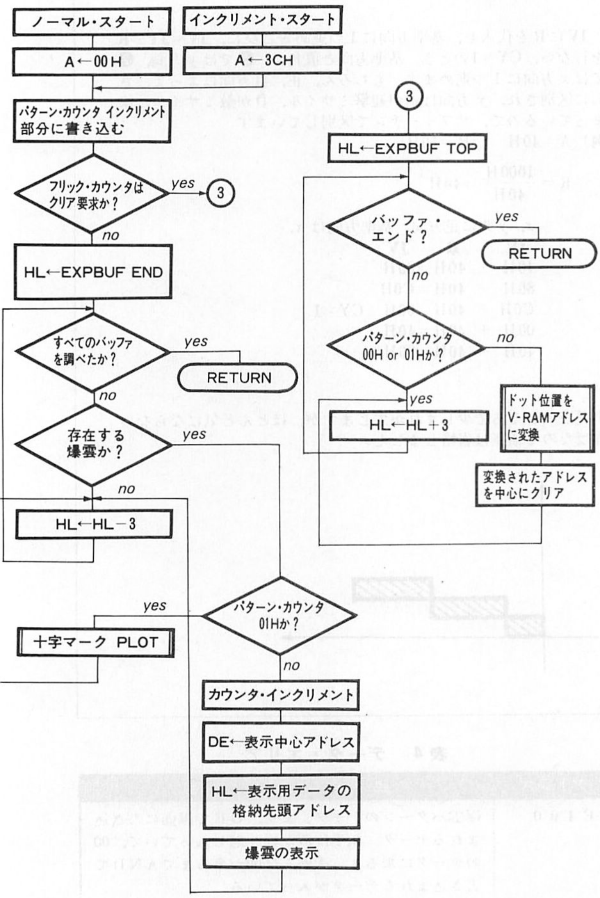
また，EXPではポインタがバッファの最後から前の方へスキャンしてきますが，これは爆雲を描くとき，その周囲を少しクリアしておき，後から生じた爆雲が前からある爆雲に重なっても，境界がわかるようにするためです．

このゲームの中で最も重要なラインのプロットについては，**図2**を見てください．N-BASICの**LINE**文はどのような処理をやっているのか知りませんが，図の方法でも正確に，しかも高速で表示できます．

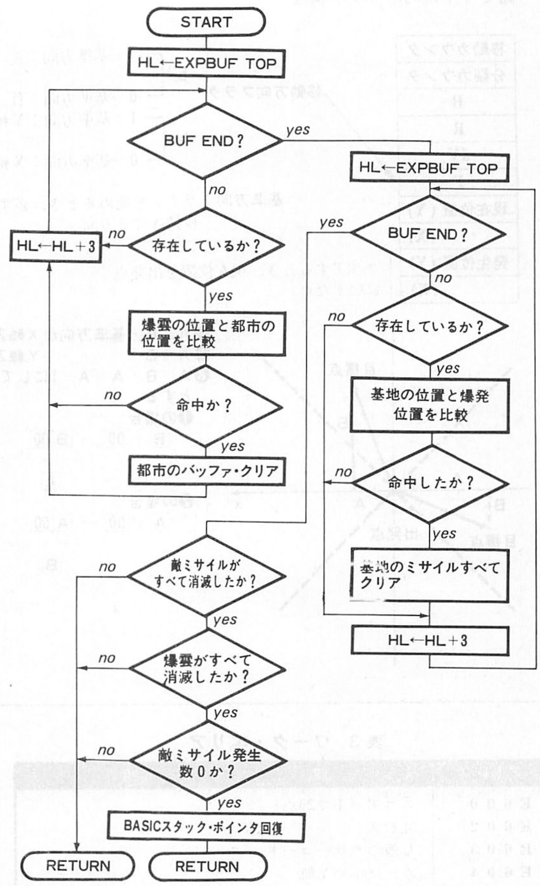
マシン語サブルーチンはD500Hから入っていますが，その詳しい説明を**表2**にあげておきます．また，ワーク・エリアについては**表3**を，データ・エリアについては**表4**を見てください．

ところでミサイル爆発の判定についてですが，敵ミサイルや破壊衛星が爆雲に触れても爆発しないことがあります．これは，命中の判定が位置の比較でなく，画面上でPOINTすることによって行なわれ，しかも1箇所のPOINTでは，ラインが交差したときやカーソルに当たっただけで爆発し

(d) 爆雲表示サブルーチン



(e) ミサイル基地・都市命中・BASICリターン



てしまうため，同時に数箇所POINTしているので，少しくらい触れただけでは命中と見なされないからです．ですから，完全に爆雲に包んでしまえば必ず爆発します．

また，ゲームを中断したいときには STOP キーを押します．これはマシン語で走っているときでも BASIC と同様に，メイン・ループで STOP キーのスキャンをしているので，即BASICへリターンし，BASIC上でBREAKがかかってくれます．

その後，もし**CONTINUE**すると，いま中断した場面のスコア計算から始まってしまいます．また，画面のリバースは，**WIDTH**， CR で元に戻ります．

プログラムの変更

ゲームのレベルは標準と思われる程度に調整してあり，なるべく変更しないで欲しいのですが，プレイヤーによっては難し過ぎるとか，何回かやって簡単になってしまったということがあるかもしれないので，変更点をあげておきます．変更と言ってもマシン語プログラムでなくBASICの**DATA**文に手を加えるだけで済みます．

データの内容は表5に示します．MOを除いて各々6つデータが並んでいるのは，2面ごとに新しいデータを使うからで，先頭が第1，2面のデータです．

色の変更は表6を参照して自由に変えて構いませんが，他の変更は少し注意が必要です．

まず，OCは都市を狙う敵ミサイル数で1以上にしてください．また，これとは別にミサイル基地を狙うものが必ず3基出現しますが，この数を変えることはできません．

反対にOCの数を9以上にしても，それ以上は発生してきません(つまり12本以上は現われない)．AOはあまり大きな数にすると，ARを大きくした場合と同じように，場面がなかなか終わらなくなります．

MOは値が小さいほどミサイルの速度が速くなり各場面ごとに変えられますが，4くらいがカーソルのスピードの関係から限界だと思います．

SAの値は小さくすると衛星が出現しやすくなり，RA

図2　ラインの表示法

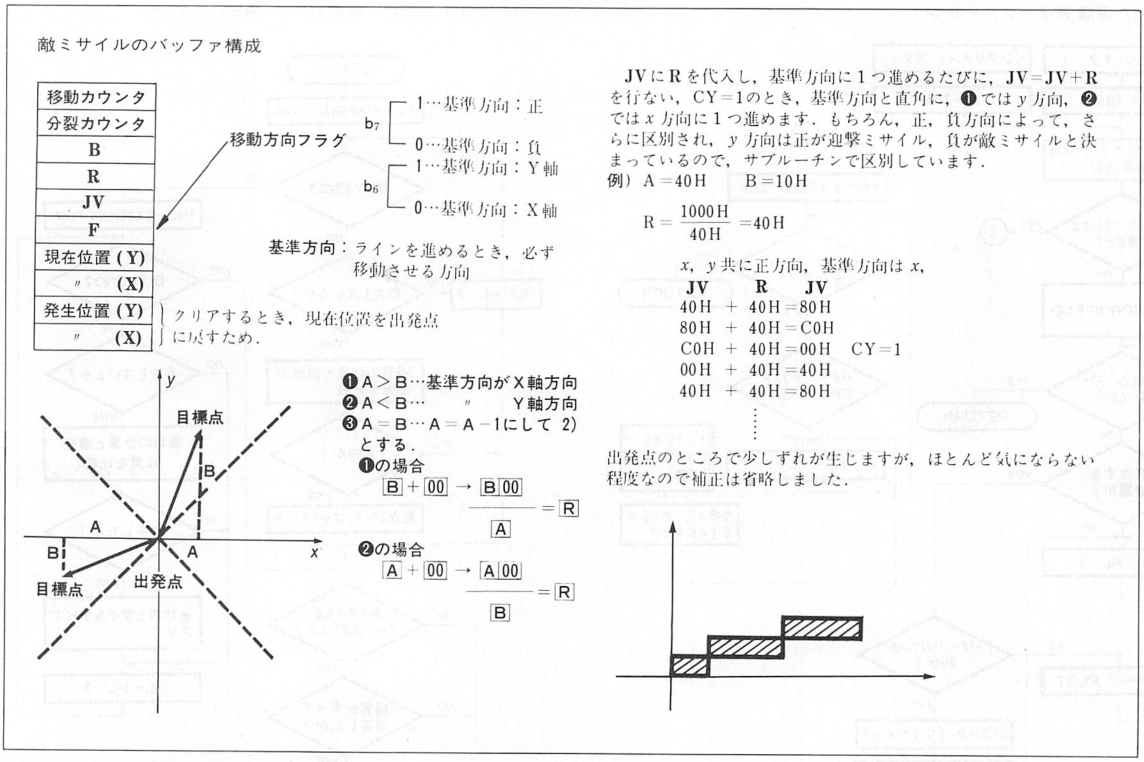


表3　ワーク・エリア

アドレス	説　明
E600	スコア（1で25点）
E602	乱数値
E603	基他のカラーコード
E604	カーソルのY値
E605	カーソルのX値
E606	敵ミサイルのカラーコード
E607	発射音発生スイッチ
E60F	敵ミサイルの速度データ
E610	都市目標の敵ミサイル数
E611	爆雲明滅カウンタ
E612	BASICスタック・ポインタ（下位）
E613	〃（上位）
E618	発射キー・スキャン・データ
E61B	分裂発生率
E61E	衛星の待機カウンタ
E61F	衛星の存在指示
E620	衛星の位置
E626	敵ミサイル　2次発生数
E627	〃　2次発生待機カウンタ
E629	〃　発生サブルーチンのカウンタ
E680～	敵ミサイルのバッファ
E710～	爆雲のバッファ
E790～	迎撃ミサイルのバッファ
E8C0～	左端ミサイル基地の各ミサイルの位置
E8E0～	中央　〃　〃
E900～	右端　〃　〃
E920～	都市の位置

表4　データ・エリア

アドレス	説　明
E100～	爆雲パターンのデータ．まず，ORで画面に書き込まれるデータ，相手位置が繰り返し入っていて，00のデータに来ると，次からは00が来るまでANDで書き込まれるデータが入っている．
E350～	クロス・マークのPLOTデータ
E360～	爆雲パターン・データの格納先頭番地
E400～	ミサイル基地の各ミサイルの位置（各場面ごとE8C0～E920へ転送される）
E460～	都市の位置（1ゲームごとE920～へ転送される）
E480～	地面表示データ
D5B5～	カーソルPLOT用データ
DDF9～	衛星表示用データ

表5　配列変数とデータの内容

配列名	行番号	内　容
C	820	敵ミサイルの色コード
CI	830	基地の色コード
OC	840	第1次ミサイル発生数
AO	850	第2次　〃
AR	860	第2次ミサイル発生間隔
MO	870	敵ミサイル速度
SA	880	破壊衛星発生間隔
RA	890	敵ミサイル分裂率

表6　カラーグラフィックと16進コード

カラー	16進コード
ブルー	38
レッド	58
マゼンタ	78
グリーン	98
シアン	B8
イエロー	D8
ホワイト	F8

の値を大きくすると敵ミサイルが分裂しやすくなります．

ディスプレイにカラーテレビやグリーン・モニタを使用している方で，青色のラインが見にくい場合はD70AHの38H(青)を別の色コードに，またリバース画面の嫌いな方は（胃に良くないかも），BASICプログラム中のOUT&H51，&H33を削ってください．



さいごに

ミサイル・コマンダーをＰＣで走らせようと思い立ってゲームの構想を考えてから，時間の合間を見ながらコツコツと作ったので，完動するまでに随分時間がかかりました．プログラム自体はあまり上手なものとは言えませんが，ミサイルの動きなど思ったよりリアルなので，我ながら感心しています．

私のハイ・スコアは24,395点ですが，ボーナス点などの配点が大きいので，初心者でも何回かプレイすれば１万点は超せると思います．

それから余談になりますが，BASICとマシン語に分れているプログラムをロードするとき，BASICのプログラムが入り終わるのを待ってからモニタでロードするわけですが，BASICが長い場合には大変じれったいものです．

そこで，自動的にモニタに入ってマシン語プログラムをロードし，しかもそのまま実行に入るようなカンタン(?)な方法を考えてみました．

方法と言っても大したことでなく，ロードする前にモニタの**S** コマンドでEA7CHから0D, 4D, 4F, 4E, 0D, 4C, 0D, 02, 52, 55, 4E, 0D([CR] **MON** [CR] **L** [CR] [S/X]**RUN** [CR])を書き込み，**cload**“ファイル名”で [RETURN] を押すかわりに [f·1] を押します．後はゲームがスタートするまで，コーヒーでも１杯．

ミサイル・コマンダー BASICプログラム・リスト

```
10 CLEAR200,&HD3FF:WIDTH40,25:CONSOLE,,0,1:OUT&H51,&H33
20 DEFINTZ,P:HI=0:DEFUSR0=&HDB20
30 DIM C(6),CI(6),OC(6),AO(6),AR(6),MO(11),SA(6),A(10),RA(6)
40 RESTORE820
50 FORI=1TO6:READC(I):NEXT
60 FORI=1TO6:READCI(I):NEXT
70 FORI=1TO6:READOC(I):NEXT
80 FORI=1TO6:READAO(I):NEXT
90 FORI=1TO6:READAR(I):NEXT
100 FORI=1TO11:READMO(I):NEXT
110 FORI=1TO6:READSA(I):NEXT
120 FORI=1TO6:READRA(I):NEXT
130 A=VAL(RIGHT$(TIME$,2))
140 POKE&HE602,A
150 GOSUB 540
160 SC=0:Z=2:POKE &HDB79,&H70:POKE&HE61D,0
170 SB=10000
180 P=Z¥2:COLOR 7:PRINT CHR$(12)
190 LOCATE10,5:PRINT "YOUR SCORE ... ";SC
200 LOCATE10,7:PRINT "HI-SCORE .... ";HI
210 LOCATE11,11:PRINT "NEXT PATTERN"
220 LOCATE10,13:PRINT P;" × POINTS"
230 FORI=1TO2000:NEXT
240 POKE&HE606,C(P):POKE&HE603,CI(P):POKE&HE610,OC(P):POKE&HE626,AO(P):POKE&HE60
F,MO(Z-1):POKE&HE61E,SA(P)
250 POKE&HE623,SA(P):POKE&HE627,AR(P):POKE&HE628,AR(P):POKE&HE600,0:POKE&HE61B,R
A(P)
260 A=USR0(0):CONSOLE,,,0:CONSOLE,,,1
270 WIDTH40:OUT&H51,&H33
280 N1=0:N2=0
290 ADD=&HE8C0:GOSUB530
300 ADD=&HE8E0:GOSUB530
310 ADD=&HE900:GOSUB530
320 ADD=&HE920
330 IFPEEK(ADD)=0THENADD=ADD+2:GOTO330:ELSEIFPEEK(ADD)=&HFFTHEN340ELSEN2=N2+1:AD
D=ADD+2:GOTO330
340 IFN2=0THEN900
350 IFN1=0THEN390
360 FORI=1TON1
370 LOCATE3,7:PRINT I*5*P:POKE&HF660+I*2,&H1E:BEEP1:FORJ=1TO10:NEXTJ:BEEP0
380 NEXTI
390 FORI=1TON2
400 LOCATE3,14:PRINT I*100*P:PUT@(11+I*9,55)-(18+I*9,59),A,PSET
410 BEEP1:FORJ=1TO10:NEXTJ:BEEP0
420 NEXTI
430 FORI=1TO1000:NEXT
440 A=PEEK(&HE600)
450 SC=SC+A*25*P+N1*5*P+N2*100*P
460 IFSC>=SBTHENSB=SB+10000:GOTO510
470 POKE&HE61D,1:POKE&HDB79,&H60
480 Z=Z+1
490 IFZ=13THENZ=2
500 GOTO180
510 D1=&HE920
520 IFPEEK(D1)=0THENPOKED1,&H60:GOTO470ELSEIFPEEK(D1)=&HFFTHEN470ELSED1=D1+2:GOT
O520
530 IFPEEK(ADD)=0THENADD=ADD+2:GOTO 530ELSEIFPEEK(ADD)=&HFFTHENRETURNELSEN1=N1+1
:ADD=ADD+2:GOTO530
540 COLOR5,0,1:PRINT CHR$(12)
550 COLOR7
560 LOCATE3,3:PRINT "*** MISSILE  COMMANDER ***"
```



I/O

ミサイル・コマンダー BASIC プログラム・リスト

```
570 LOCATE5,5:PRINT "HI-SCORE..";HI;"  SCORE..";SC
580 LINE(3,7)-(36,22)," ",7,BF
590 LOCATE7,7:PRINT "* POINTS *"
600 LOCATE7,11:PRINT "* BONUS *"
610 LOCATE5,9:PRINT "KILLER SATELLITE..  100Pt."
620 LOCATE5,10:PRINT "ATTACK MISSILE  ..   25Pt."
630 LOCATE5,13:PRINT "REMAINED MISSILE..    5Pt."
640 LOCATE5,14:PRINT "DEFENCED CITY   ..100Pt."
650 RESTORE810
660 X=62:Y=55
670 FORI=1TO24
680 READ A :IFA=0THENY=Y+1:GOTO 710
690 A=X+A
700 PSET(A,Y)
710 NEXT
720 GET@(63,55)-(70,59),A,G
730 LOCATE10,15:PRINT "SHOOT KEY         cursor"
740 LOCATE10,17:PRINT "┌─┴─┐              8"
750 LOCATE8,18:PRINT "□□□□ □            │"
760 LOCATE8,19:PRINT "□□□□ □          4─┼─6"
770 LOCATE7,20:PRINT "└────┘              │"
780 LOCATE7,21:PRINT "                    2"
790 LOCATE5,22:PRINT "GAME START‥<Y>"
800 IFINKEY$="y"THENRETURNELSE800
810 DATA 3,6,0,1,3,4,5,6,0,1,2,3,4,5,6,7,0,1,2,3,4,5,6,7
820 DATA &H58,&H98,&Hd8,&Hb8,&H58,&Hd8
830 DATA &Hd8,&Hd8,&H78,&H78,&Hf8,&Hf8
840 DATA 5,4,3,2,3,1
850 DATA 5,7,5,6,6,5
860 DATA &H20,&H20,&H40,&H50,&H68,&H50
870 DATA 9,9,7,6,6,6,7,6,6,5,4
880 DATA &Ha0,&H80,&H78,&H78,&H68,&H78
890 DATA &H10,&H20,&H30,&H30,&H30,&H30
900 A=PEEK(&HE600)
910 SC=SC+A*25*P+N1*5*P
920 IFSC>=SBTHENSB=SB+10000:GOTO 510
930 WIDTH40,25:COLOR2:PRINT CHR$(12)
940 LOCATE8,9:PRINT "█████ █     █ ████"
950 LOCATE8,10:PRINT "█        ██   █ █   █"
960 LOCATE8,11:PRINT "█        █ █  █ █    █"
970 LOCATE8,12:PRINT "████     █  █ █ █    █"
980 LOCATE8,13:PRINT "█        █   ██ █    █"
990 LOCATE8,14:PRINT "█        █    █ █   █"
1000 LOCATE8,15:PRINT "█████ █     █ ████"
1010 FORI=1TO30
1020 BEEP1:OUT&H51,&H30:FORJ=1TO10:NEXTJ:BEEP0
1030  OUT&H51,&H33:FORJ=1TO10:NEXTJ
1040 NEXTI
1050 IFSC>HITHENHI=SC
1060 GOTO 150
```



ミサイル・コマンダー マシン語ダンプ・リスト

```
Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D500 E5 C5 D5 32 5B EA 7C FE A0 D2 20 D5 7D FE 64 D2 :A88
D510 20 D5 CD 24 D5 B6 77 2A 63 EA 3A 5B EA CD F8 04 :8A7
D520 D1 C1 E1 C9 7D E6 03 57 3E FC A5 0F 0F 6F CB 0C :83C
D530 DA 44 D5 22 63 EA D5 CD F3 03 D1 3E 01 14 15 C8 :7FB
D540 07 C3 3E D5 3E 7F A4 67 22 63 EA D5 CD F3 03 D1 :87D
D550 3E 10 C3 3D D5 E5 C5 D5 7C FE 9F D2 6A D5 7D FE :A47
D560 64 D2 6A D5 CD 24 D5 2F A6 77 D1 C1 E1 C9 E5 C5 :A6D
D570 D5 7D FE 9F D2 81 D5 7D FE 64 D2 81 D5 CD 24 D5 :AE4
D580 A6 D1 C1 E1 C9 7E FE 80 CA 9F D5 5F 23 56 23 EB :A02
D590 7C 80 67 7D 81 6F 3E F8 CD 00 D5 EB C3 85 D5 23 :8D3
D5A0 7E FE 80 C8 5F 23 56 EB 7C 80 67 7D 81 6F CD 55 :879
D5B0 D5 EB C3 9F D5 80 00 00 00 01 00 02 01 00 02 00 :47D
D5C0 FF 00 FE 00 00 FF 00 FE 80 80 ED 4B 04 E6 21 B5 :7F2
D5D0 D5 CD 85 D5 DB 00 CB 57 C2 E3 D5 0C 3E 4E B9 D2 :996
D5E0 E3 D5 0D DB 00 CB 67 C2 F2 D5 05 3E 05 B8 DA F2 :927
D5F0 D5 04 DB 00 CB 77 C2 01 D6 04 3E 98 B8 D2 01 D6 :7CA

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D600 05 DB 01 CB 47 C2 10 D6 0D 3E 06 B9 DA 10 D6 0C :671
D610 21 B6 D5 CD 85 D5 ED 43 04 E6 C9 E5 D5 5E 23 56 :947
D620 23 4E 23 46 72 2B 73 2B 70 2B 71 2B 2B 2B 7E 23 :443
D630 77 2B 2B E5 CD 66 D8 56 2B 5E EB CD 55 D5 EB 23 :88C
D640 23 7E BB E1 30 ED D1 E1 C9 00 00 00 00 00 00 00 :5D5
D650 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
D660 F5 CD 36 10 F1 3D C2 60 D6 C9 C5 3A 02 E6 47 87 :8AC
D670 87 80 3C 32 02 E6 C1 C9 21 80 E4 11 D8 FC 01 68 :7BA
D680 01 ED B0 06 50 21 40 FE 36 FF 23 10 FB 36 00 23 :60F
D690 36 D8 23 36 50 C3 B6 D6 7E A7 CA B1 D6 3D 5F 23 :83B
D6A0 56 E5 EB CD 24 D5 D1 36 38 2B 36 3F EB 23 10 E8 :7D1
D6B0 C9 23 23 10 E3 C9 06 0B 21 C0 E8 CD 98 D6 06 09 :6EF
D6C0 21 E0 E8 CD 98 D6 06 0A 21 00 E9 CD 98 D6 21 20 :7BA
D6D0 E9 7E 3C C8 3D 28 1A 5F 23 56 E5 EB CD 24 D5 D1 :829
D6E0 36 FE 2B 36 FF 2B 36 FE 23 23 23 36 CE EB 23 18 :686
D6F0 E0 23 23 18 DC 21 90 E7 7E 3C CA 75 D7 3D C2 51 :7D2

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D700 D7 23 CD 0E D8 56 2B 5E EB 3E 38 CD 00 D5 EB 01 :77B
D710 05 00 09 35 CA 1E D7 01 06 00 09 C3 F8 D6 23 7E :544
D720 2B 77 2B 46 2B 4E 2B 2B 70 2B 71 2B 2B 2B 7E 23 :410
D730 77 2B 2B 2B 36 01 23 E5 CD 0E D8 56 2B 5E EB CD :681
D740 55 D5 EB 01 05 00 09 7E 3D 77 CA 65 D7 E1 C3 37 :737
D750 D7 01 10 00 09 C3 F8 D6 23 23 23 C3 BE D7 01 10 :654
D760 00 09 C3 D3 D7 C1 23 23 5E 23 56 EB 36 02 EB 23 :685
D770 23 23 C3 F8 D6 21 18 E6 DB 07 BE C8 77 CB 6F CA :8D9
D780 8D D7 CB 77 CA 93 D7 CB 7F CA 99 D7 C9 21 C0 E8 :AF0
D790 C3 9C D7 21 E0 E8 C3 9C D7 21 00 E9 7E 3C C8 3D :91E
D7A0 C2 A8 D7 23 23 C3 9C D7 5F 36 00 23 56 3E 01 32 :63C
D7B0 07 E6 2A 04 E6 EB E5 D5 CD BE D8 21 11 E7 7E A7 :947
D7C0 C2 58 D7 E5 C1 36 01 D1 22 0D E6 23 73 23 72 D5 :7B4
D7D0 21 90 E7 7E 3C C2 E0 D7 2A 0D E6 36 00 E1 E1 C9 :8A9
D7E0 3D CA 5E D7 36 00 23 EB 21 08 E6 01 04 00 ED B0 :631
D7F0 EB D1 C1 71 23 70 23 23 71 23 70 23 3A 0C E6 77 :691

Add   +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
D800 23 77 23 ED 5B 0D E6 73 23 72 CD 78 D6 C9 C5 D5 :87E
D810 23 4E 23 56 23 7E 07 DA 40 D8 07 DA 2E D8 7A 81 :666
D820 30 03 23 35 2B 2B 77 23 23 23 35 D1 C1 C9 7A 81 :54C
D830 30 05 23 23 35 2B 2B 2B 77 23 23 35 23 D1 C1 C9 :4A1
D840 07 DA 54 D8 7A 81 30 03 23 35 2B 2B 77 23 23 23 :469
D850 34 D1 C1 C9 7A 81 30 05 23 23 34 2B 2B 2B 77 23 :554
D860 23 35 23 D1 C1 C9 C5 D5 23 4E 23 56 23 7E 07 DA :6DC
D870 98 D8 07 DA 86 D8 7A 81 30 03 23 34 2B 2B 77 23 :624
D880 23 23 35 D1 C1 C9 7A 81 30 05 23 23 35 2B 2B 2B :502
D890 77 23 23 34 23 D1 C1 C9 07 DA AC D8 7A 81 30 03 :702
D8A0 23 34 2B 2B 77 23 23 23 34 D1 C1 C9 7A 81 30 05 :54C
D8B0 23 23 34 2B 2B 2B 77 23 23 34 23 D1 C1 C9 7C 92 :578
```

ミサイル・コマンダー　マシン語ダンプ・リスト

D8C0	F5	1F	E6	80	47	F1	30	02	ED	44	32	08	E6	7D	93	30	:775
D8D0	02	ED	44	32	09	E6	4F	3A	08	E6	91	30	0F	3A	08	E6	:5C3
D8E0	32	0A	E6	79	32	0C	E6	3E	40	B0	18	0B	3A	08	E6	32	:56A
D8F0	0C	E6	79	32	0A	E6	78	32	0B	E6	3A	08	E6	4F	3A	09	:5E2

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
D900	E6	B9	20	03	3D	18	FA	38	11	61	2E	00	45	4F	CD	2A	:574
D910	D9	32	09	E6	32	0A	E6	C9	00	00	67	2E	00	45	CD	2A	:5B6
D920	D9	32	09	E6	32	0A	E6	C9	00	00	AF	37	3F	ED	42	D8	:711
D930	3C	18	F8	21	80	E6	7E	3C	C8	3D	20	11	01	0A	00	09	:4D7
D940	18	F4	01	06	00	09	5E	23	56	2B	EB	18	1D	3D	77	20	:412
D950	F1	3A	0F	E6	77	23	7E	3D	CA	DA	D9	77	23	CD	66	D8	:897
D960	56	2B	5E	EB	3A	06	E6	CD	00	D5	CD	6E	D5	28	32	2C	:728
D970	CD	6E	D5	28	2C	25	CD	6E	D5	28	26	2D	CD	6E	D5	28	:74C
D980	20	2D	CD	6E	D5	28	1A	24	CD	6E	D5	28	14	24	CD	6E	:66E
D990	D5	28	0E	2C	CD	6E	D5	28	08	2C	CD	6E	D5	28	02	18	:5F5
D9A0	08	EB	23	23	23	23	C3	36	D9	25	2D	EB	CD	1B	D6	E5	:731
D9B0	2B	2B	2B	2B	2B	2B	36	00	3E	52	BB	38	07	3A	00	E6	:3E2
D9C0	3C	32	00	E6	21	11	E7	7E	A6	28	05	23	23	23	18	F7	:536
D9D0	36	02	23	73	23	72	E1	C3	A2	D9	36	F0	E5	01	05	00	:693
D9E0	09	5E	23	56	EB	22	16	E6	CD	6A	D6	E6	01	3C	47	21	:681
D9F0	80	E6	C5	7E	3C	CA	36	DA	3D	28	06	01	0A	00	09	18	:556

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
DA00	F2	3A	0F	E6	77	23	36	F0	23	E5	1E	6C	CD	6A	D6	E6	:866
DA10	7F	C6	10	57	2A	16	E6	CD	BE	D8	D1	21	08	E6	01	04	:71A
DA20	00	ED	B0	2A	16	E6	EB	73	23	72	23	73	23	72	23	C1	:6C5
DA30	10	C0	E1	C3	5C	D9	C1	E1	C3	5C	D9	AF	32	79	DA	18	:98F
DA40	05	3E	3C	32	79	DA	3A	11	E6	A7	CA	D7	DA	AF	32	11	:749
DA50	E6	21	68	E7	7E	3C	C8	3D	20	05	2B	2B	2B	18	F5	47	:60F
DA60	FE	01	20	14	23	4E	23	46	E5	21	50	E3	CD	85	D5	E1	:74E
DA70	2B	2B	2B	2B	2B	C3	54	DA	00	3C	FE	0E	20	08	E5	C5	:5E2
DA80	CD	78	D6	C1	E1	AF	77	FE	03	20	07	E5	C5	CD	A7	DF	:A08
DA90	C1	E1	E5	78	87	C6	5C	6F	26	E3	56	23	5E	E1	23	4E	:849
DAA0	23	46	2B	2B	E5	C5	E1	D5	CD	24	D5	D1	EB	7E	A7	28	:8EE
DAB0	0D	23	46	23	4E	23	EB	E5	09	B6	77	E1	18	EE	23	7E	:698
DAC0	A7	28	0D	23	46	23	4E	EB	E5	09	A6	77	E1	EB	18	EE	:77E
DAD0	E1	2B	2B	2B	C3	54	DA	3C	32	11	E6	21	11	E7	7E	3C	:68B
DAE0	C8	3D	E6	FE	20	05	23	23	23	18	F3	23	5E	23	56	EB	:667
DAF0	D5	CD	24	D5	01	74	00	09	0E	09	11	46	00	EB	ED	B0	:60F

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
DB00	EB	01	7F	FF	09	01	0A	00	EB	ED	B0	EB	01	7E	FF	09	:778
DB10	01	09	00	EB	ED	B0	EB	E1	23	C3	DE	DA	00	00	00	00	:6FC
DB20	21	00	00	39	22	12	E6	31	F0	D4	01	19	50	CD	3A	09	:4E3
DB30	01	FF	00	CD	F7	08	21	30	50	22	04	E6	3E	05	32	1C	:50A
DB40	E6	AF	32	1F	E6	3E	02	32	24	E6	32	25	E6	3A	03	E6	:6A8
DB50	32	D1	E4	32	91	D6	32	49	E5	06	07	21	C1	E5	77	23	:74E
DB60	23	23	23	10	F9	3E	03	32	29	E6	3E	05	32	15	E6	32	:496
DB70	1A	E6	21	00	E4	11	C0	E8	01	60	00	ED	B0	21	0D	E7	:6D1
DB80	06	80	36	FF	23	10	FB	21	11	E7	06	1E	36	00	23	23	:4A2
DB90	23	10	F9	21	90	E7	11	10	00	06	0A	36	01	19	10	FB	:450
DBA0	36	FF	21	80	E6	11	0A	00	06	0C	36	00	19	10	FB	36	:479
DBB0	FF	3E	01	21	64	0A	CD	92	DC	3E	01	21	64	53	CD	92	:67E
DBC0	DC	3E	01	21	64	96	CD	92	DC	3A	10	E6	47	CD	D2	DB	:862
DBD0	18	23	21	20	E9	0E	00	7E	3C	28	15	3D	20	04	23	23	:311
DBE0	18	F5	5F	0C	23	56	EB	3E	01	CD	92	DC	EB	10	F0	C9	:80A
DBF0	79	A7	C8	18	DD	21	50	F3	06	16	11	76	00	36	00	23	:53D

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
DC00	36	F8	23	36	50	19	10	F5	CD	78	D6	3E	20	CD	60	D6	:771
DC10	0E	06	06	08	21	40	40	CD	92	DF	E5	C5	CD	CA	D5	C1	:7D8
DC20	E1	2D	2D	2D	24	24	24	10	EE	0D	20	E6	3A	25	E6	3D	:567
DC30	32	25	E6	20	09	3A	24	E6	32	25	E6	CD	CA	D5	CD	F5	:815
DC40	D6	CD	33	D9	3A	1A	E6	3D	32	1A	E6	20	0C	3A	15	E6	:6B9
DC50	32	1A	E6	CD	41	DA	C3	5C	DC	CD	3B	DA	CD	04	DD	00	:8A5
DC60	00	00	3E	01	00	00	00	CD	60	D6	CD	18	DE	DB	09	CB	:5B4
DC70	47	CA	E4	DD	CD	55	DF	3A	07	E6	A7	CA	2C	DC	AF	32	:954
DC80	07	E6	3E	20	D3	40	CD	36	10	CD	36	10	AF	D3	40	C3	:709
DC90	2C	DC	D5	C5	F5	01	0A	00	E5	21	80	E6	7E	3C	20	05	:6ED
DCA0	E1	F1	C1	D1	C9	3D	28	03	09	18	F1	3A	0F	E6	77	CD	:81A
DCB0	6A	D6	47	3A	1B	E6	B8	38	11	CD	6A	D6	E6	1F	47	3E	:75A
DCC0	10	80	23	77	00	00	00	00	18	03	23	36	FA	23	D1	D5	:461
DCD0	E5	CD	F8	DC	67	2E	08	E5	CD	BE	D8	E1	D1	E5	21	08	:A2B
DCE0	E6	01	04	00	ED	B0	E1	EB	73	23	72	23	73	23	72	E1	:768
DCF0	F1	3D	C2	94	DC	C1	D1	C9	CD	6A	D6	FE	90	38	02	C6	:A56

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
DD00	80	C6	08	C9	21	11	E7	06	00	7E	3C	CA	66	DD	3D	20	:65A
DD10	05	23	23	23	18	F3	23	7E	FE	5E	38	F6	23	7E	FE	87	:6CA
DD20	30	F1	FE	7A	38	06	78	32	2A	E9	18	E7	FE	6A	38	06	:739
DD30	78	32	28	E9	18	DD	FE	5C	38	06	78	32	26	E9	18	D3	:6EC
DD40	FE	46	30	CF	FE	38	38	07	78	32	24	E9	C3	13	DD	FE	:820
DD50	28	38	07	78	32	22	E9	C3	13	DD	FE	1C	DA	13	DD	78	:72B
DD60	32	20	E9	C3	13	DD	21	11	E7	7E	3C	28	4C	3D	20	05	:597
DD70	23	23	23	18	F4	23	5E	23	56	CD	7F	DD	23	18	EA	7B	:638
DD80	FE	5A	D8	E5	7A	FE	8A	38	0C	21	00	E9	06	14	36	00	:6B5
DD90	23	10	FB	E1	C9	FE	58	30	FA	FE	4C	38	0C	21	E0	E8	:8CF
DDA0	06	12	36	00	23	10	FB	E1	C9	FE	12	30	E6	21	C0	E8	:715
DDB0	06	16	36	00	23	10	FB	E1	C9	01	0A	00	21	80	E6	7E	:53A
DDC0	3C	28	05	3D	C0	09	18	F7	21	11	E7	7E	3C	CA	D7	DD	:6CF
DDD0	3D	C0	23	23	23	18	F4	3A	26	E6	A7	C0	18	06	00	00	:53D
DDE0	00	00	00	00	2A	12	E6	F9	C9	00	00	00	00	00	00	00	:2E4
DDF0	00	00	00	00	00	00	00	00	80	00	00	00	01	00	00	01	:082

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
DE00	01	01	02	02	03	03	02	FF	03	FE	FF	FF	00	FE	FE	FE	:706
DE10	FF	02	00	03	FE	03	80	80	3A	1C	E6	3D	32	1C	E6	C2	:674
DE20	47	DF	3E	05	32	1C	E6	3A	1D	E6	3D	C0	3A	1F	E6	3D	:653
DE30	28	33	3A	1E	E6	3D	32	1E	E6	C0	3C	32	1F	E6	CD	6A	:676
DE40	D6	FE	80	38	0C	3E	02	32	21	E6	3E	06	32	20	E6	18	:5A5
DE50	0A	3E	FE	32	21	E6	3E	9A	32	20	E6	CD	6A	D6	E6	7F	:801
DE60	C6	10	32	22	E6	21	F9	DD	3A	20	E6	47	0E	26	C5	CD	:754
DE70	85	D5	C1	3A	21	E6	80	FE	99	30	04	FE	07	30	0B	AF	:796
DE80	32	1F	E6	3A	23	E6	32	1E	E6	C9	32	20	E6	47	21	FA	:713
DE90	DD	C5	CD	85	D5	E1	25	CD	6E	D5	28	52	2C	CD	6E	D5	:995
DEA0	28	4C	2C	24	CD	6E	D5	28	45	24	CD	6E	D5	28	3F	2D	:609
DEB0	24	CD	6E	D5	28	38	2D	CD	6E	D5	28	32	2D	25	CD	6E	:6B8
DEC0	D5	28	2B	25	CD	6E	D5	28	25	3A	00	E6	C6	04	32	00	:5C6
DED0	E6	21	11	E7	7E	3C	CA	7F	DE	3D	28	05	23	23	23	18	:5CB
DEE0	F3	36	02	23	36	26	23	3A	20	E6	77	C3	7F	DE	3A	26	:604
DEF0	E6	A7	C8	3A	22	E6	3D	32	22	E6	C0	3E	F0	32	22	E6	:836

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
DF00	00	01	0A	00	21	80	E6	7E	3C	C8	3D	28	03	09	18	F7	:494
DF10	3A	0F	E6	77	CD	6A	D6	E6	1F	47	3E	05	80	23	77	23	:67F
DF20	CD	F8	DC	57	1E	6C	E5	3A	20	E6	67	2E	26	CD	BE	D8	:8C5
DF30	D1	21	08	E6	01	04	00	ED	B0	EB	36	26	23	3A	20	E6	:62C
DF40	77	23	36	26	23	77	C9	3A	1F	E6	A7	C8	3A	20	E6	67	:6AE
DF50	2E	26	C3	96	DE	3A	26	E6	A7	C8	3A	29	E6	3D	32	29	:721
DF60	E6	C0	3E	03	32	29	E6	3A	28	E6	3D	32	28	E6	C0	CD	:77A
DF70	6A	D6	E6	2F	47	3A	27	E6	80	32	28	E6	CD	6A	D6	E6	:896
DF80	03	28	F9	47	3A	26	E6	90	30	01	AF	32	26	E6	CD	D2	:6FE
DF90	DB	C9	E5	3E	20	D3	40	CD	AE	DF	AF	D3	40	CD	AE	DF	:A70
DFA0	25	20	F0	E1	C9	00	00	21	20	20	CD	92	DF	C9	7D	3D	:701
DFB0	20	FD	C9	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:ED9
DFC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
DFD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
DFE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
DFF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
E000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
E010	00	00	00	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:BF4
E020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:FF0
E030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:FF0
E040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
E050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
E060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
E070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
E080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:FF0
E090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:FF0
E0A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:FF0
E0B0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	:FF0
E0C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
E0D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
E0E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
E0F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
E100	66	00	00	06	00	01	60	FF	FF	00	F9	00	01	9F	FF	FF	:662
E110	00	66	00	00	9F	00	01	F9	FF	FF	00	00	00	00	00	00	:3FD
E120	FF	00	00	77	00	01	02	00	02	77	FF	FF	20	FF	FE	88	:695
E130	FF	88	00	77	00	01	DA	00	02	77	FF	FF	AD	FF	FE	EE	:8E8
E140	00	78	BB	FF	88	77	FF	89	77	FF	87	00	00	00	00	00	:6B6
E150	FF	00	00	FF	00	01	6F	00	02	FF	FF	FF	F6	FF	FE	11	:871
E160	00	78	01	00	79	10	00	77	88	FF	88	08	FF	89	80	FF	:697
E170	87	00	6F	00	02	F9	00	03	F6	FF	FE	9F	FF	FD	DD	00	:85F
E180	78	ED	00	79	FE	00	7A	DE	00	77	EF	00	76	BB	FF	88	:852
E190	7B	FF	89	F7	FF	8A	B7	FF	87	7F	FF	86	00	00	00	00	:8C4
E1A0	00	00	00	00	00	00	00	00	FF	00	00	FF	00	01	FF	00	:2FE
E1B0	02	6F	00	03	FF	FF	FF	FF	FF	FE	F6	FF	FD	33	00	78	:A0A
E1C0	13	00	79	01	00	7A	31	00	77	10	00	76	CC	FF	88	8C	:514
E1D0	FF	89	08	FF	8A	C8	FF	87	80	FF	86	00	BB	00	78	DB	:97A
E1E0	00	79	ED	00	7A	FE	00	7B	6F	00	03	F6	FF	FD	BD	00	:77A
E1F0	77	DE	00	76	EF	00	75	DD	FF	88	BD	FF	89	7B	FF	8A	:9DC

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
E200	F7	FF	8B	DB	FF	87	B7	FF	86	7F	FF	85	00	00	00	00	:921
E210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:000
E220	FF	00	00	FF	00	01	FF	00	02	FF	00	03	6F	00	04	FF	:574
E230	FF	FF	FF	FF	FE	FF	FF	FD	F6	FF	FC	77	00	78	37	00	:C0C
E240	79	13	00	7A	01	00	7B	73	00	77	31	00	76	10	00	75	:398
E250	EE	FF	88	CE	FF	89	8C	FF	8A	08	FF	8B	EC	FF	87	C8	:BAC
E260	FF	86	80	FF	85	00	77	00	78	B7	00	79	DB	00	7A	ED	:7EA
E270	00	7B	FE	00	7C	7B	00	77	BD	00	76	DE	00	75	EF	00	:65C
E280	74	6F	00	04	F6	FF	FC	EE	FF	88	DE	FF	89	BD	FF	8A	:AF9
E290	7B	FF	8B	F7	FF	8C	ED	FF	87	DB	FF	86	B7	FF	85	7F	:C14
E2A0	FF	84	00	00	00	00	00	00	00	00	00	00	00	00	00	00	:183
E2B0	FF	00	00	FF	00	01	FF	00	02	FF	00	03	FF	00	04	FF	:604
E2C0	FF	FF	FF	FF	FE	FF	FF	FD	FF	FF	FC	77	00	78	77	00	:C55
E2D0	79	33	00	7A	13	00	7B	01	00	7C	77	00	77	33	00	76	:3C8
E2E0	31	00	75	10	00	74	EE	FF	88	EE	FF	89	CC	FF	8A	8C	:8F6
E2F0	FF	8B	08	FF	8C	EE	FF	87	CC	FF	86	C8	FF	85	80	FF	:BAD

Add	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	Sum
E300	84	00	77	00	78	77	00	79	BB	00	7A	DB	00	7B	ED	00	:5DB
E310	7C	77	00	77	BB	00	76	BD	00	75	DE	00	74	EE	FF	85	:791
E320	EE	FF	89	DD	FF	8A	BD	FF	8B	7B	FF	8C	EE	FF	87	DD	:C7A

ミサイル・コマンダー マシン語ダンプ・リスト

```
E330 FF 86 DB FF 85 B7 FF 84 00 00 00 00 00 00 00 00 :61E
E340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E350 00 00 01 01 01 FF FF 01 FF FF 80 80 00 00 00 00 :500
E360 E1 00 E1 20 E1 50 E1 A8 E2 20 E2 B0 E2 B0 E2 20 :9C4
E370 E1 A8 E1 50 E1 20 E3 90 00 00 00 00 00 00 00 00 :52E
E380 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E390 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E3A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E3B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E3C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E3D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E3E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E3F0 00 00 00 00 00 00 00 00 00 00 C7 C7 C7 C7 C7 C7 :4AA

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E400 5D 06 5D 0A 5D 0E 61 04 61 08 61 0C 61 10 65 06 :34C
E410 65 0A 65 0E 65 12 FF FF FF FF FF FF FF FF FF FF :B4F
E420 5D 50 5D 54 61 4E 61 52 61 56 65 4C 65 50 65 54 :596
E430 65 58 FF FF FF FF FF FF FF FF FF FF FF FF FF FF :EAF
E440 5D 92 5D 96 5D 9A 61 90 61 94 61 98 65 8E 65 92 :7A2
E450 65 96 65 9A FF FF FF FF FF FF FF FF FF FF FF FF :DEE
E460 60 20 60 30 60 40 60 62 60 72 60 82 FF FF FF FF :822
E470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E480 88 88 CC CC C8 CC 8C C8 0C 00 00 00 00 00 00 00 :59C
E490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E4A0 00 00 00 00 00 00 00 88 CC 88 8C 00 00 00 00 00 :268
E4B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
```

```
E4C0 00 00 00 00 00 00 00 C0 0C C8 CC 8C CC CC 88 88 :594
E4D0 00 D8 50 00 00 00 00 00 00 00 00 00 00 00 00 00 :128
E4E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :000
E4F0 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF :7F8

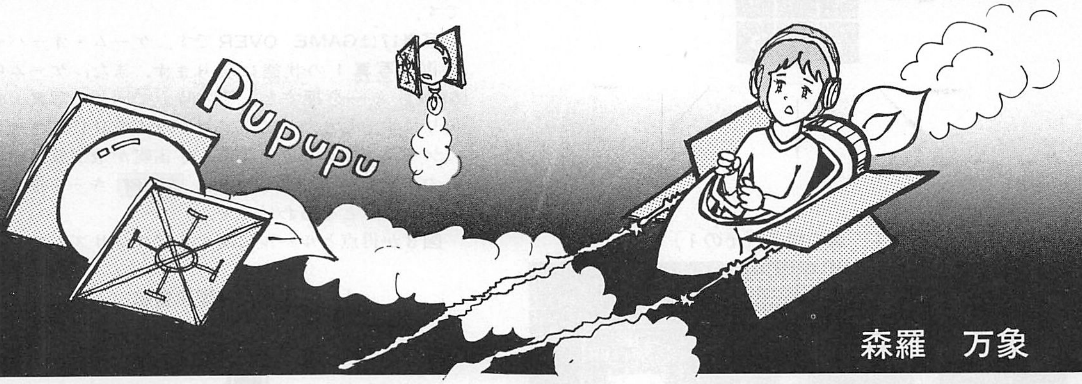
Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E500 FF 8C 88 00 00 00 00 00 00 00 00 00 00 00 00 00 :213
E510 00 00 00 00 00 00 00 00 00 00 00 00 00 80 F8 FF :277
E520 FF FF FF FF 8F 00 00 00 00 00 00 00 00 00 00 00 :48B
E530 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0E FF :10D
E540 FF FF FF FF FF FF FF FF 00 D8 50 E8 50 E8 50 E8 :C78
E550 50 E8 50 E8 50 E8 50 E8 50 E8 50 E8 50 E8 50 E8 :9C0
E560 50 E8 50 E8 50 E8 50 E8 50 E8 50 E8 50 E8 50 00 :8D8
E570 FF FF FF FF FF FF FF FF FF FF FF 0F 00 CC 00 00 :BD0
E580 00 00 CC 00 00 CC 00 00 00 00 CC 00 00 CC 00 00 :330
E590 00 00 CC 00 C0 FC FF FF FF FF FF FF FF 00 CC 00 :A4D
E5A0 00 00 00 CC 00 00 CC 00 00 00 00 CC 00 00 CC 00 :330
E5B0 00 00 00 CC 00 E0 FF FF FF FF FF FF FF FF FF FF :BA2
E5C0 00 D8 0E B8 12 D8 16 B8 1A D8 1E B8 22 D8 2F B8 :6FF
E5D0 33 D8 37 B8 3B D8 3F B8 43 D8 50 00 00 00 00 00 :56F
E5E0 00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FF :7F8
E5F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :FF0

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
E600 04 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF :EF5

Add  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  Sum
```

PC-8001 宇宙追撃ゲーム

スペース・チェイス



森羅　万象

遠いむかし，はるか数万光年かなたの銀河系で，とてつもない大冒険が繰り広げられた……*

そして物語も後半，帝国軍の巨大要塞惑星デス・スターを破壊するため，惑星ヤヴィンの第4衛星にある共和国軍の秘密基地から，次々とX型宇宙艇が飛びたった．目標はデス・スターの弱点，それは巨大な溝の底にある，要塞惑星全体に電力を供給している発電所の主反応炉へ直通する放熱口だ！　(STAR WARS より)

*この部分は映画“STAR WARS”の日本版ポスター第1号から引用したものですが，たとえば我が銀河系の直径が10万光年（円板部の有効直径）で，アンドロメダ銀河までは220万光年なので，この『はるか数万光年かなたの…』はおかしいと思うのですが….

ゲームの説明

RUNさせると，タイトルが表示され（写真1），中央から周囲へ広がる四角形によりこのタイトルが消されていきます（写真2）．

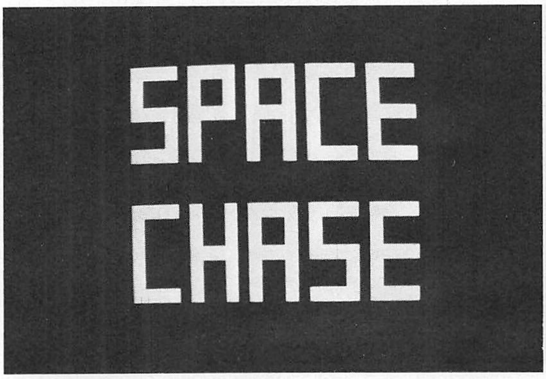


写真1　タイトルが表示され…

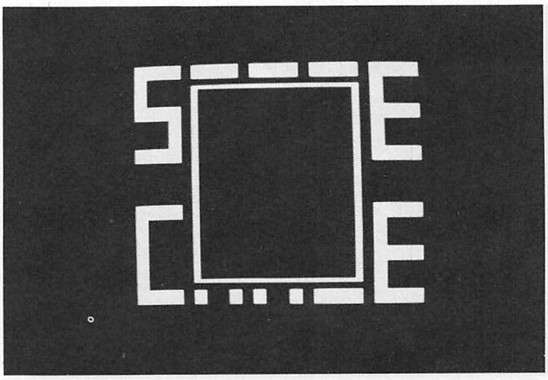


写真2　次にタイトルが序々に消されていきます．

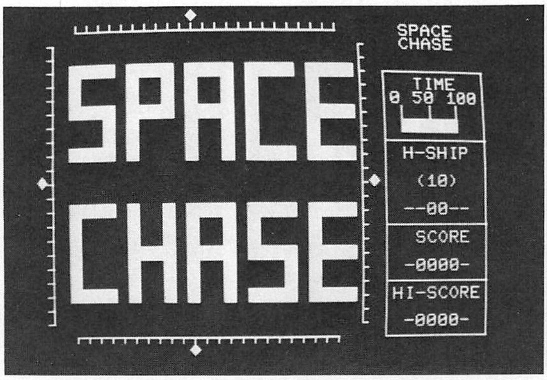


写真3　そして，ゲーム・スタート！

そして再びタイトルと目盛，照準，スコアなどが表示され（写真3），いよいよゲーム・スタートです．

写真4のように溝とH型宇宙艇が表示されるので，テン・キーにより照準を上下左右に動かしてH型宇宙艇に合わせます（図1）．そして，スペース・キーを押すとレーザーが発射されます．溝のパターンは写真4，5，6と3種類あり，これがこの順序で順番に繰り返し表示されるので，

図1　キー入力と照準の移動方向

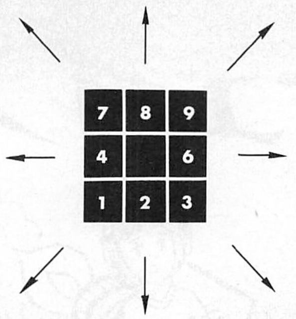


写真4　溝の様子（その1）

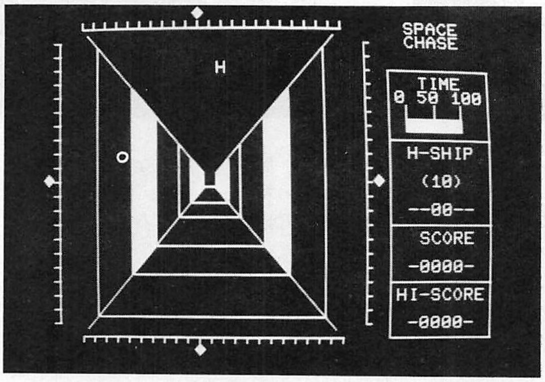


写真5　溝の様子（その2）

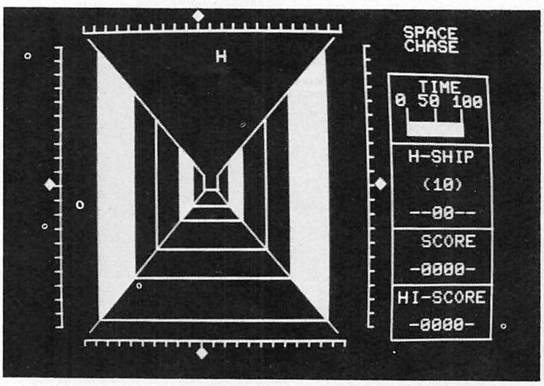
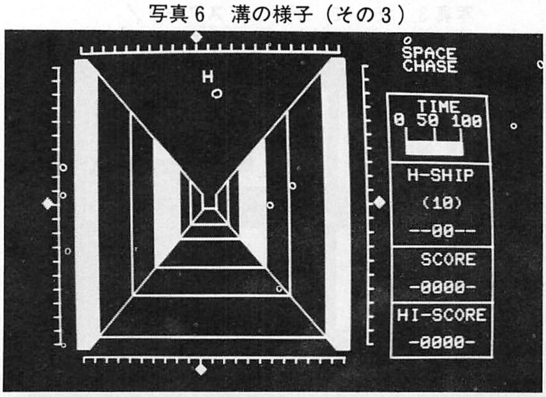


写真6　溝の様子（その3）



自分が前方に進んでいるように感じられます．H型宇宙艇には上の写真の大きさの他に，**写真7**，**8**と合計3種類があります（**図2**）．**写真9**はレーザーが命中して爆発したところ，また**写真10**は外れたところです．

また，時々放熱口が接近してきます（**写真11〜15**）．この写真ではH型宇宙艇が映っていませんが，実際には常に表示されています．**写真16**はこれを攻撃して爆発したところです．

写真17はGAME　OVERです．ゲーム・オーバーとなれば再び**写真1**の状態に戻ります．また，ゲーム中などに **SHIFT** キーを押すと，その時点で同じく**写真1**の状態に戻ります．したがって，プログラムをRUNさせてもすぐにゲームを始めなくてよく，宇宙艇が飛び放熱口が現われるのを見ながら，おもむろに **SHIFT** キーを押せばゲーム・スタートとなるわけです．

図3が得点とルールに関するアルゴリズムです．H型宇

図2　H型宇宙艇

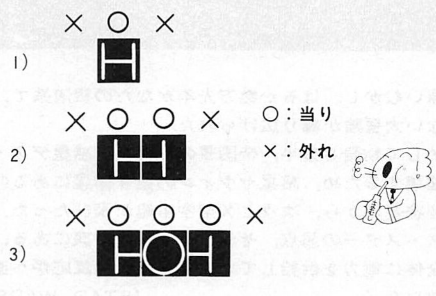


写真7　H型宇宙艇（2）

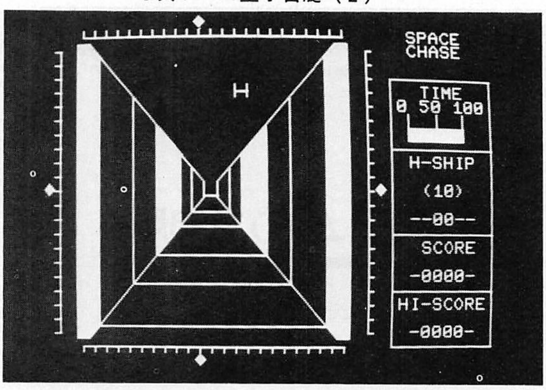


写真8　H型宇宙艇（3）

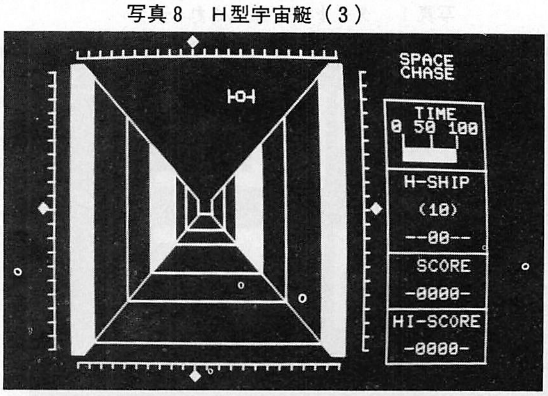


写真９　宇宙艇に命中！

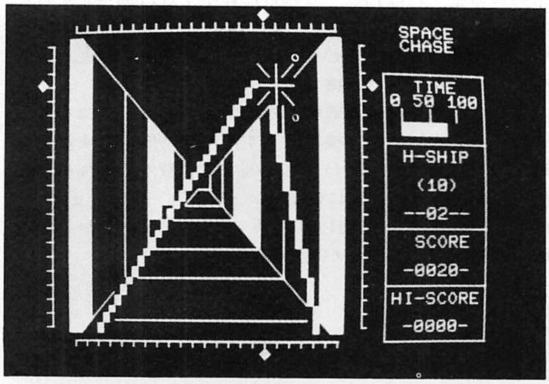


写真10　ハズレ！

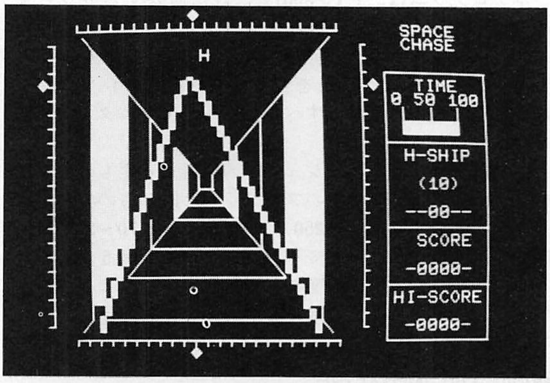


写真11　放熱口接近（その１）

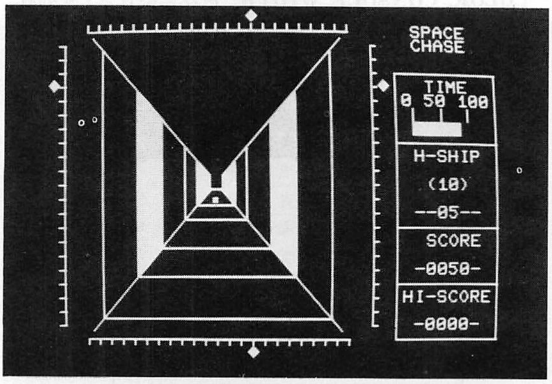


写真12　放熱口接近（その２）

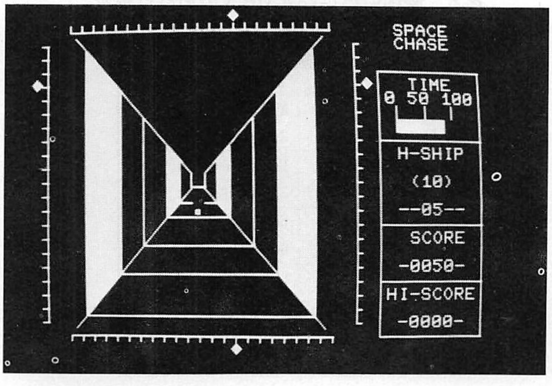


写真13　放熱口接近（その３）

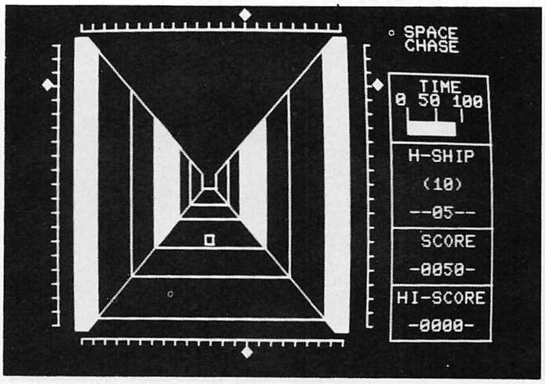


写真14　放熱口接近（その４）

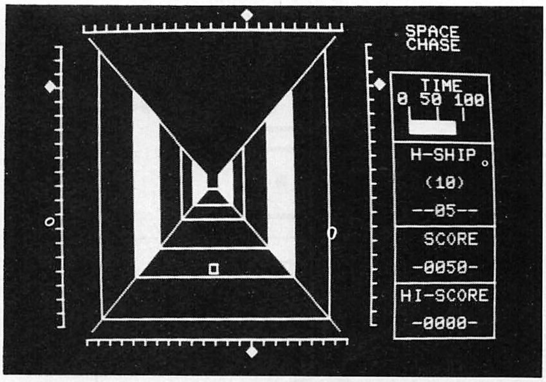


写真15　放熱口接近（その５）

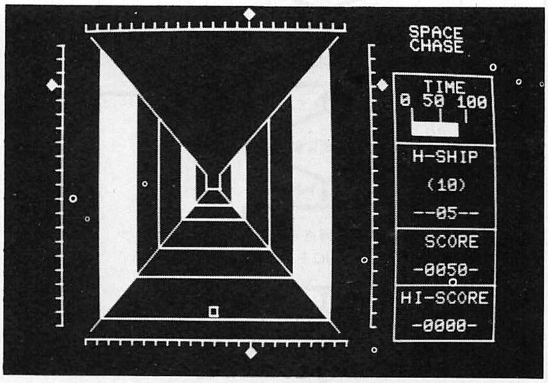


写真16　放熱口に命中！

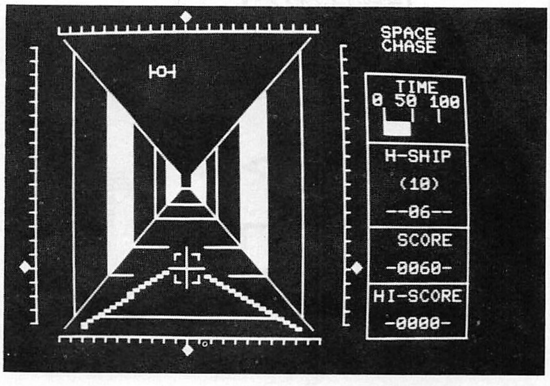


写真17　ゲーム・オーバー

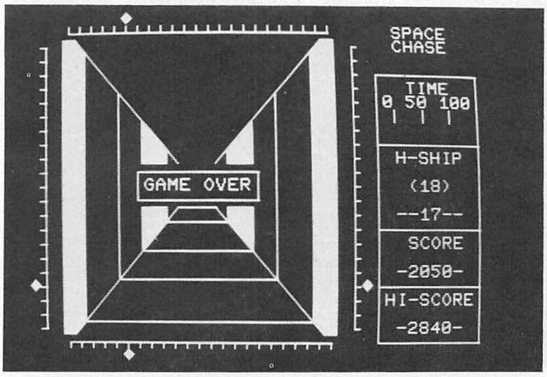
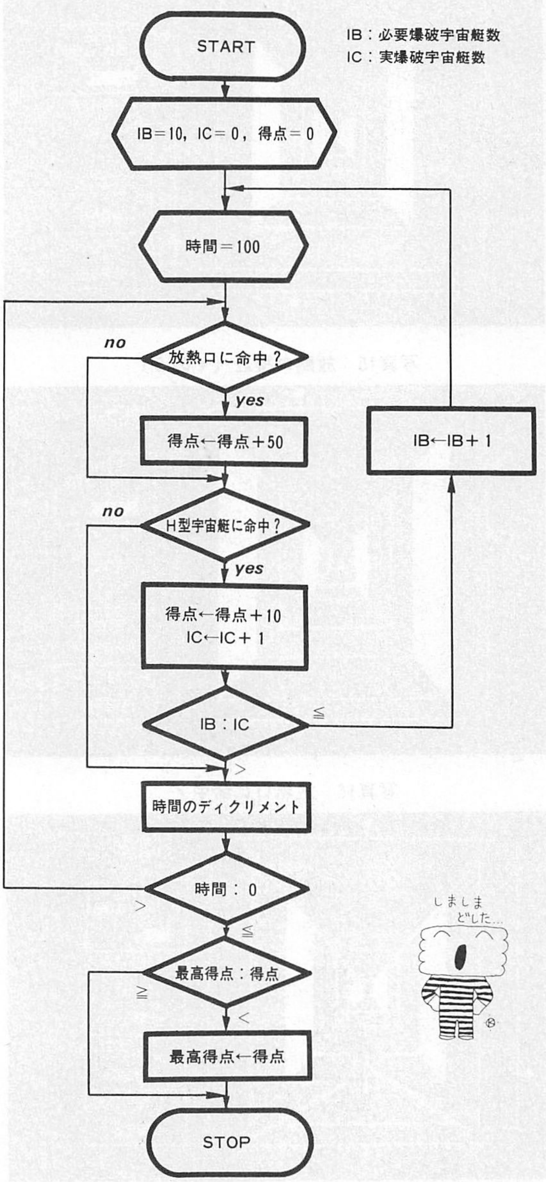


図3　得点とルールに関するフローチャート



宙艇に命中すれば10点，放熱口に命中すれば50点がそれぞれ得られます．そして一定時間内に爆破しなければならない宇宙艇数，すなわち必要爆破宇宙艇数があらかじめ決まっています．最初は10機です．

そして，10機全部を破壊すると時間が満タン！になり，必要爆破宇宙艇数が１機増えます．一定時間内に実際に爆破した宇宙艇数，すなわち実爆破宇宙艇数が必要爆破宇宙艇数に到達しない場合にはゲーム・オーバーとなります．画面右側のスコアーの“H-SHIP”の欄の（　）内が必要爆破宇宙艇数で，その下が実爆破宇宙艇数です．

プログラムの説明

本プログラムは，ＰＣ-8001とこれに適合するカラーモニタ用のN-BASICで作られています．もちろん，グリーン・モニタでもそのまま使用できますが，色の違いが濃淡差となります．色や濃淡を変える場合には**表１，２**を参考にしてください．写真はすべてグリーン・モニタで同じ濃さで撮映したものです．

図４がプログラム・モジュールの構造を示すもので，各モジュール名は**リスト１**のプログラムと同じものです．プリンタの都合上，行番号250，260行および1570～2190行に関しては**写真18～21**を見てください．また，**図５**が画面レイアウトで，**表３**が主な変数の一覧表です．

では，以下に行番号に従ってプログラムを簡単に説明していきます．

❶120～170　INITIALIZE

COLOR, CONSOLE, WIDTH，それに型宣言と配列宣言などです．また最高得点を０点に初期設定し，ユーザー

表1 〝COLOR〟ファンクション・コードと色，濃淡の関係

ファンクション・コード	カラーモニタ	グリーン・モニタ
0	黒	淡
1	青	↑
2	赤	
3	紫	
4	緑	
5	水色	
6	黄	↓
7	白	濃

表２　画面の色とCOLOR文のある行番号

部　分	ファンクション・コード	COLOR文のある行番号
タイトル "SPACE CHASE"	6	1400
タイトルを消して行く四角形	4	290
目盛	6	310
照準	6	1260, 1300, 1340, 1370
スコア	5	350, 640, 670, 740
溝	4	610, 620, 630, 1120
H型宇宙艇	7	960
放熱口	6	820
レーザー光線	3	1070
爆発パターン	3, 7	1140, 1160, 1200, 1220
"GAME OVER"	6	680

図4　プログラム・モジュールの構造

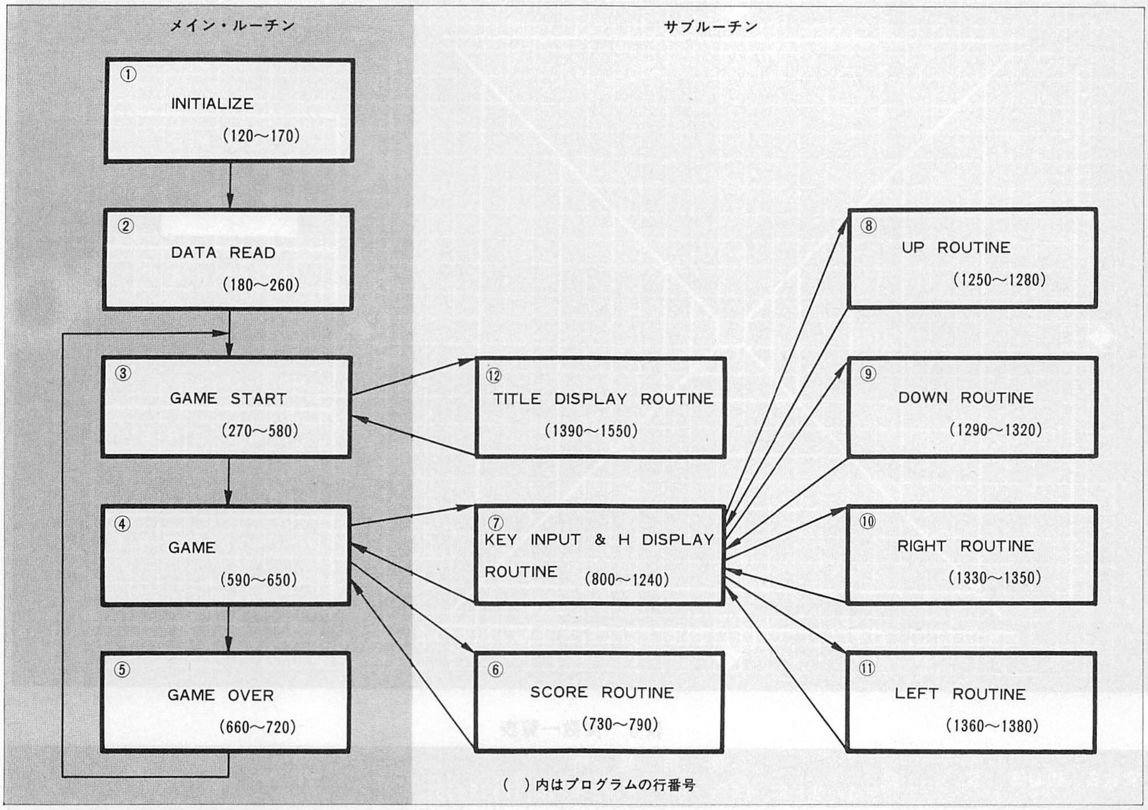


写真18　250, 260行のリスト

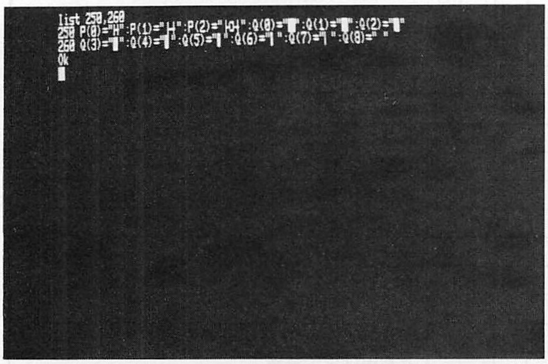


写真20　1780～1980行のリスト

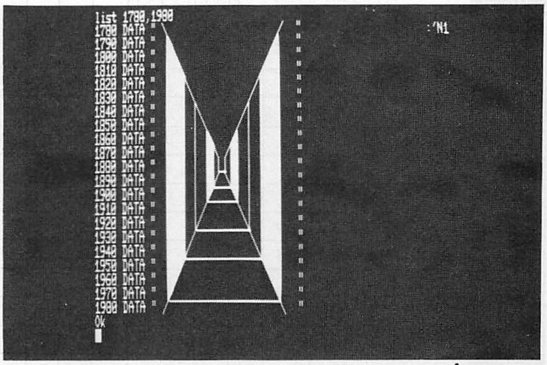


写真19　1570～1770行のリスト

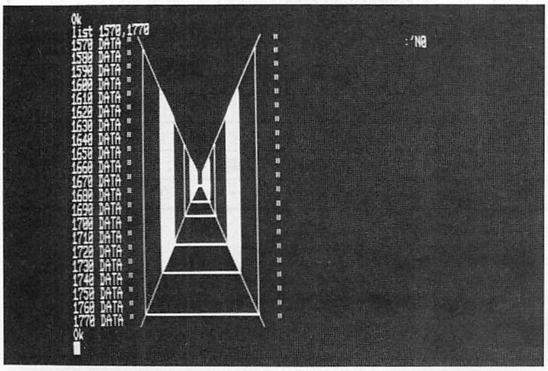


写真21　1990～2190行のリスト

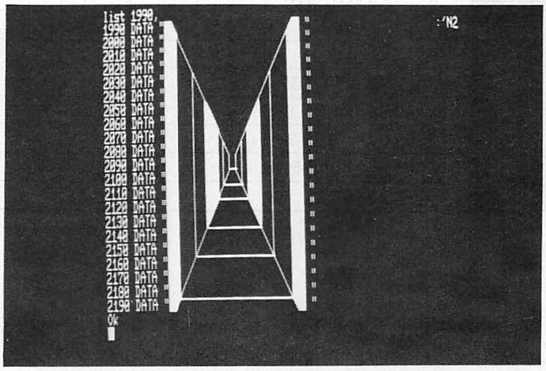


図5　画面レイアウト

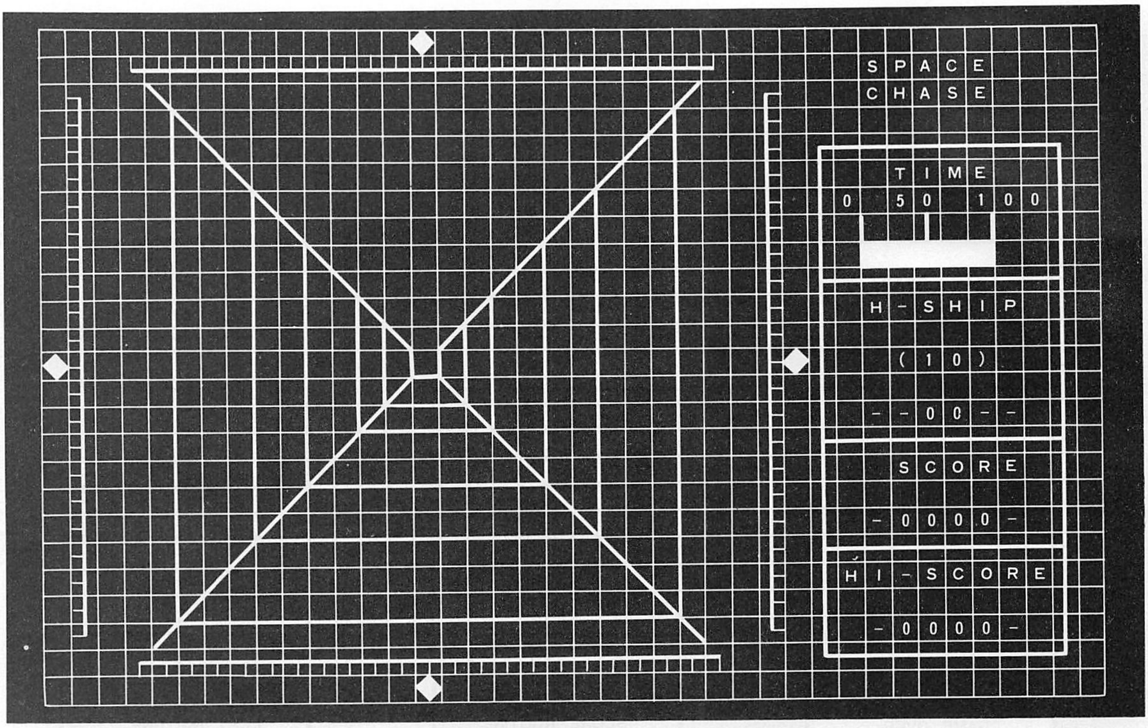


表3　変数一覧表

変数名	初期値	変域	使用場所 1	2	3	4	5	6	7	8	9	10	11	12	13	内容
A(0)			○						○							H型宇宙艇を表示するときの背景のストア用エリア
N0(252)			○	○		○									○	溝のPUT@用データ
N1(252)			○	○		○									○	同　上
N2(252)			○	○		○									○	同　上
P(2)			○	○					○							H型宇宙艇表示用データ
Q(8)			○	○		○										"TIME" のバーグラフ表示用データ
IX	12	3≦IX≦25	○		○				○			○	○			照準のX位置(キャラクタ単位)
IY	13	2≦IY≦22	○		○				○	○	○					照準のY位置(　同　上　)
JX	乱数により決定	4≦JX≦24	○		○				○							H型宇宙艇のX位置(　同　上　)
JY	乱数により決定	3≦JY≦21	○		○				○							H型宇宙艇のY位置(　同　上　)
KX		6≦KX≦50	○						○							照準のX位置(グラフィック単位)
KY		8≦KY≦88	○						○							照準のY位置(　同　上　)
IE	0	0≦IE　5	○		○				○							放熱口の位置に関するパラメータ
LY	0	13, 14, 16, 18, 21	○		○				○							放熱口のY位置(キャラクタ単位)
IZ		JX+IJ/2−1	○						○							爆発パターンのX位置(キャラクタ単位)
IB	10	10≦IB	○		○			○								必要爆破宇宙艇数
IC	0	0≦IC≦IB	○		○			○								実爆破宇宙艇数
IS	0	0≦IS	○		○		○	○								得点
IH	0	0≦IH	○		○		○									最高得点
IT	35	31≦IT≦35	○			○		○								時間に関するパラメータ
JT	0	0≦JT≦8	○			○		○								同　上
IJ	0	0, 1, 2	○		○											H型宇宙艇の大きさを示すパラメータ
ID	0	0, 10, 50	○		○			○								0：外れ, 10：H型宇宙艇に命中, 50：放熱口に命中
IA	0	0, 1	○		○	○		○								0：ゲーム・オーバー, 1：ゲーム続行

使用場所の番号は，図1の番号やプログラムの説明の行番号の頭の番号と同じです．

関数**FNZ**を定義しています．この関数**FNZ(I, J)**は整変数Iの頭に0を付け加えてJ桁の数字とするものです．たとえば，

I＝47　　J＝4

とすれば，FNZ(I, J)は，

```
        0047
```

となります．

❷180〜260　DATA　READ

配列N０，N１，N２への **DATA** 文（1560〜2190行）よりのREADと，配列P，Qへの代入です．ただし，N０，N１，N２へはそのまま代入するのではなく，**DATA**文の文字列を２字ずつに区切りASCII符号に変換して代入します．

これはN０，N１，N２が**PUT**＠でモニタへ表示させるためのデータだからです（250，260行に関しては**写真18**を見てください）．

❸270〜580　GAME　START

各種パラメータの初期設定，タイトルの表示，そして目盛，照準とスコアの表示を行ないます．タイトルの表示は，実際には⓬で行ないます．

❹590〜650　GAME

溝の表示と“TIME”のバーグラフの表示を行ない，❻，❼へGOSUBします．

❺660〜720　GAME　OVER

ゲーム終了時の処理で，“GAME　OVER”の表示と最高得点の表示を行ないます．そして一定時間後に❸へジャンプします．

❻730〜790　SCORE　ROUTINE

H型宇宙艇または放熱口に命中したときの得点の加算，表示，それに必要爆破宇宙艇数と実爆破宇宙艇数との比較を行ない，その結果によりゲーム・オーバーとなったり必要爆破宇宙艇数がインクリメントされたりします．

❼800〜1240　KEY　INPUT　&　H　DISPLAY　ROUTINE

●810〜880
放熱口の表示をするかどうかの判断と，実際の表示をします．
●890〜960
H型宇宙艇の移動と大きさの変更を行ない，表示します．
●970〜1060
キー入力の受け付けを行ない，その内容により分岐します．オート・リピートとするため，**INP**命令によりキー入力の受け付けを行なっています．本プログラムで使用しているキーが押された場合のポートの変化を**表4**に示します．
●1070〜1120
レーザーの発射と命中かどうかの判断です．
●1130〜1180
H型宇宙艇にレーザーが命中した場合の爆発パターンの表示と爆発音の出力です．
●1190〜1240
放熱口にレーザーが命中した場合の爆発パターンの表示と爆発音の出力です．

表4　キー入力とポートの変化

キー	ポート番号	変化内容(FF→)
1	0	FD
2	0	FB
3	0	F7
4	0	EF
6	0	BF
7	0	7F
8	1	FE
9	1	FD
スペース	9	BF
SHIFT	8	BF

❽1250〜1280　UP　ROUTINE

照準の上方への移動の表示です．

❾1290〜1320　DOWN　ROUTINE

照準の下方への移動の表示です．

❿1330〜1350　RIGHT　ROUTINE

照準の右方への移動の表示です．

⓫1360〜1380　LEFT　ROUTINE

照準の左方への移動の表示です．

⓬1390〜1550　TITLE　DISPLAY　ROUTINE

タイトル"SPACE　CHASE"の表示です．

⓭1560〜2190　DATA　DEFINITION

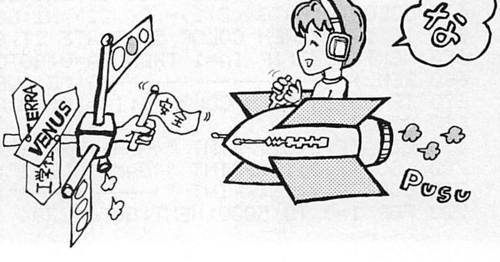
N０，N１，N２用のデータです（この部分に関しては**写真19〜21**を見てください）．

おわりに

高得点を上げるコツをあなただけにそっとお教えします．まず，（必要爆破宇宙艇数－１）機を破壊します．このときにはたとえ放熱口が接近してきても無視します．

上記の数だけ破壊すれば，今度は放熱口を破壊するために照準を合わせ固定しておきます．そして残り時間がわずかになれば残りの１機を破壊します．また，照準の移動はレーザー発射よりも優先されるので，H型宇宙艇を破壊するときにはスペース・キーは左手で押したままにしておき，右手でテン・キーをあやつり，照準をH型宇宙艇に合わせます．

そして，合えば右手をテン・キーから離します．すると，間髪を入れず，レーザーが発射されるわけです．また，斜め方向に移動するときには，[1]，[3]，[7]，[9]のキーを使います．[4]，[6]と[2]，[8]で移動するよりも早く移動できます．



スペース・チェイス　プログラム・リスト

```
10 REM ********************************************
20 REM *                                          *
30 REM *              SPACE CHASE  Ver 1.0        *
40 REM *                                          *
50 REM *        for  NEC PC-8001 BASIC Ver 1.0    *
60 REM *                                          *
70 REM ********************************************
80 REM *       Copyright (C) 1980  by Ohashi M.   *
90 REM ********************************************
100 REM
110 REM ***===============MAIN  ROUTINE==============***
120 REM *-----------------INITIALIZE------------------*
130 COLOR 7,32,0:CONSOLE 0,25,0,1:WIDTH 40,25
140 DEFSNG A-H:DEFINT I-N:DEFSTR O-Z
150 DEFFNZ(I,J)=RIGHT$("000"+MID$(STR$(I),2),J)
160 DIM A(0),N0(252),N1(252),N2(252),P(2),Q(8)
170 IH=0
180 REM *-----------------DATA READ-------------------*
190 FOR I=1 TO 21:READ Z:FOR J=1 TO 12:X=MID$(Z,J*2,1):Y=MID$(Z,J*2-1,1)
200 N0(I*12+J-13)=VAL("&H"+HEX$(ASC(X))+HEX$(ASC(Y))):NEXT:NEXT
210 FOR I=1 TO 21:READ Z:FOR J=1 TO 12:X=MID$(Z,J*2,1):Y=MID$(Z,J*2-1,1)
220 N1(I*12+J-13)=VAL("&H"+HEX$(ASC(X))+HEX$(ASC(Y))):NEXT:NEXT
230 FOR I=1 TO 21:READ Z:FOR J=1 TO 12:X=MID$(Z,J*2,1):Y=MID$(Z,J*2-1,1)
240 N2(I*12+J-13)=VAL("&H"+HEX$(ASC(X))+HEX$(ASC(Y))):NEXT:NEXT
250 P(0)="H":P(1)=" H":P(2)=" ├┤":Q(0)=" ":Q(1)=" ":Q(2)=" "
260 Q(3)=" ":Q(4)=" ":Q(5)=" ":Q(6)=" ":Q(7)=" ":Q(8)=" "
270 REM *-----------------GAME START------------------*
280 PRINT CHR$(&HC):GOSUB 1390:IJ=0:IA=0:IB=10:IC=0:ID=0:IE=0:IS=0:IX=12:IY=13
290 COLOR 4:FOR I=0 TO 24:I0=29-I:I1=48-I*2:I2=30+I:I3=51+I*2
300 LINE (I0,I1)-(I2,I3),PSET,B:LINE (I0,I1)-(I2,I3),PRESET,B:NEXT
310 LY=0:JX=9+RND(1)*10:JY=4+RND(1)*7:GOSUB 1250:GOSUB 1330:COLOR 6
320 LOCATE 3,1 :PRINT "┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴┴";
330 LOCATE 3,23:PRINT "┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬┬";
340 LOCATE 1,2:PRINT "┐";SPC(25);"┌";:LOCATE 1,22:PRINT "┘";SPC(25);"└";
350 FOR I=3 TO 21:LOCATE 1,I:PRINT "┤";SPC(25);"├";:NEXT:COLOR 5
360 LOCATE 29,1 :PRINT "  SPACE   ";
370 LOCATE 29,2 :PRINT "  CHASE   ";
380 LOCATE 29,4 :PRINT "┌────────┐";
390 LOCATE 29,5 :PRINT "│  TIME  │";
400 LOCATE 29,6 :PRINT "│0 50 100│";
410 LOCATE 29,7 :PRINT "││ │  │ │";
420 LOCATE 29,8 :PRINT "│ ▒▒▒▒▒  │";
430 LOCATE 29,9 :PRINT "├────────┤";
440 LOCATE 29,10:PRINT "│ H-SHIP │";
450 LOCATE 29,11:PRINT "│        │";
460 LOCATE 29,12:PRINT "│  (10)  │";
470 LOCATE 29,13:PRINT "│        │";
480 LOCATE 29,14:PRINT "│ --00-- │";
490 LOCATE 29,15:PRINT "├────────┤";
500 LOCATE 29,16:PRINT "│  SCORE │";
510 LOCATE 29,17:PRINT "│        │";
520 LOCATE 29,18:PRINT "│ -0000- │";
530 LOCATE 29,19:PRINT "├────────┤";
540 LOCATE 29,20:PRINT "│HI-SCORE│";
550 LOCATE 29,21:PRINT "│        │";
560 LOCATE 29,22:PRINT "│ -";FNZ(IH,4);"- │";
570 LOCATE 29,23:PRINT "└────────┘";
580 GOSUB 1390
590 REM *-------------------GAME--------------------*
600 FOR IT=35 TO 31 STEP -1:FOR JT=0 TO 8
610 COLOR 4:PUT@ (3,2)-(26,22),N0:GOSUB 800:GOSUB 730
620 COLOR 4:PUT@ (3,2)-(26,22),N1:GOSUB 800:GOSUB 730
630 COLOR 4:PUT@ (3,2)-(26,22),N2:GOSUB 800:GOSUB 730
640 IF IA=0 THEN COLOR 5:LOCATE IT,8:PRINT Q(JT);
650 NEXT:NEXT:IF IA=1 THEN IA=0:GOTO 590
660 REM *-----------------GAME OVER-------------------*
670 IF IS>IH THEN COLOR 5:IH=IS:LOCATE 32,22:PRINT FNZ(IH,4)
680 COLOR 6
690 LOCATE 9,11:PRINT "┌─────────┐";
700 LOCATE 9,12:PRINT "│GAME OVER│";
710 LOCATE 9,13:PRINT "└─────────┘";
720 FOR I=0 TO 5000:NEXT:GOTO 270
```

```
730 REM ***===============SCORE ROUTINE===============***
740 IF ID=0 THEN RETURN ELSE COLOR 5
750 IF ID=10 THEN IC=IC+1:LOCATE 33,14:PRINT FNZ(IC,2)
760 IS=IS+ID:ID=0:LOCATE 32,18:PRINT FNZ(IS,4)
770 IF IC<IB THEN RETURN
780 IB=IB+1:IC=0:LOCATE 33,14:PRINT "00":LOCATE 31,8:PRINT "░░░░";
790 LOCATE 33,12:PRINT USING "##";IB;:IT=31:JT=8:IA=1:RETURN
800 REM ***=======KEY INPUT & H DISPLAY ROUTINE=======***
810 IF IE=0 THEN IF RND(1)>.05 THEN LY=0:GOTO 890 ELSE IE=1
820 COLOR 6:ON IE GOTO 830,840,850,860,870
830 LOCATE 14,13:PRINT "。":LY=13:GOTO 880
840 LOCATE 14,14:PRINT "。":LY=14:GOTO 880
850 LOCATE 14,16:PRINT "□":LY=16:GOTO 880
860 LOCATE 14,18:PRINT "□":LY=18:GOTO 880
870 LOCATE 14,21:PRINT "□":LY=21
880 IF IE=5 THEN IE=0 ELSE IE=IE+1
890 BEEP1:BEEP0:FOR J=0 TO 2:B=RND(1):C=RND(1)
900 IF B<.2 THEN IF JX>4  THEN JX=JX-1:GOTO 940 ELSE GOTO 940
910 IF B<.4 THEN IF JX<24 THEN JX=JX+1:GOTO 940 ELSE GOTO 940
920 IF B<.6 THEN IF JY>3  THEN JY=JY-1:GOTO 940 ELSE GOTO 940
930 IF B<.8 THEN IF JY<21 THEN JY=JY+1
940 IF C<.15 THEN IF IJ>0 THEN IJ=IJ-1
950 IF C>.85 THEN IF IJ<2 THEN IJ=IJ+1
960 GET@ (JX,JY)-(JX+2,JY),A:COLOR 7:LOCATE JX,JY:PRINT P(IJ)
970 FOR I=0 TO 100:NEXT:I0=INP(0):I1=INP(1):I8=INP(8):I9=INP(9)
980 IF I1=&HFE THEN GOSUB 1250:GOTO 1120
990 IF I0=&HFB THEN GOSUB 1290:GOTO 1120
1000 IF I0=&HBF THEN GOSUB 1330:GOTO 1120
1010 IF I0=&HEF THEN GOSUB 1360:GOTO 1120
1020 IF I1=&HFD THEN GOSUB 1250:GOSUB 1330:GOTO 1120
1030 IF I0=&HF7 THEN GOSUB 1290:GOSUB 1330:GOTO 1120
1040 IF I0=&H7F THEN GOSUB 1250:GOSUB 1360:GOTO 1120
1050 IF I0=&HFD THEN GOSUB 1290:GOSUB 1360:GOTO 1120
1060 IF I8=&HBF THEN IT=31:JT=8:RETURN
1070 IF I9<>&HBF GOTO 1120 ELSE KX=IX*2:KY=IY*4:COLOR 3:BEEP1
1080 LINE (11,91)-(KX,KY),PSET  :LINE (47,91)-(KX,KY),PSET
1090 LINE (11,91)-(KX,KY),PRESET:LINE (47,91)-(KX,KY),PRESET
1100 BEEP0:IF JX<=IX AND IX<=JX+IJ AND IY=JY GOTO 1130
1110 IF IX=14 AND IY=LY GOTO 1190
1120 COLOR 4:PUT@ (JX,JY)-(JX+2,JY),A:NEXT:RETURN
1130 FOR I=0 TO 5:IZ=JX+IJ/2-1
1140 COLOR 3:LOCATE IZ,IY-1:PRINT "\|/";:LOCATE IZ,IY:PRINT "─┼─";
1150 LOCATE IZ,IY+1:PRINT "/|\";:FOR J=0 TO 10:BEEP1:BEEP0:NEXT
1160 COLOR 7:LOCATE IZ,IY-1:PRINT "/|\";:LOCATE IZ,IY:PRINT "-×-";
1170 LOCATE IZ,IY+1:PRINT "\|/";:FOR J=0 TO 10:BEEP1:BEEP0:NEXT
1180 NEXT:ID=10:JX=9+RND(1)*10:JY=4+RND(1)*7:RETURN
1190 FOR I=0 TO 5:IZ=IX-1
1200 COLOR 3:LOCATE IZ,IY-1:PRINT "┌|┐";:LOCATE IZ,IY:PRINT "─┼─";
1210 LOCATE IZ,IY+1:PRINT "└|┘";:FOR J=0 TO 10:BEEP1:BEEP0:NEXT
1220 COLOR 7:LOCATE IZ,IY-1:PRINT "┘|└";:LOCATE IZ,IY:PRINT "-×-";
1230 LOCATE IZ,IY+1:PRINT "┐|┌";:FOR J=0 TO 10:BEEP1:BEEP0:NEXT
1240 NEXT:ID=50:LY=0:IE=0:RETURN
1250 REM ***================UP ROUTINE================***
1260 IF IY=2 THEN RETURN ELSE COLOR 6
1270 LOCATE 0,IY:PRINT " ";:LOCATE 28,IY:PRINT " ";:IY=IY-1
1280 LOCATE 0,IY:PRINT "◆";:LOCATE 28,IY:PRINT "◆";:RETURN
1290 REM ***===============DOWN ROUTINE===============***
1300 IF IY=22 THEN RETURN ELSE COLOR 6
1310 LOCATE 0,IY:PRINT " ";:LOCATE 28,IY:PRINT " ";:IY=IY+1
1320 LOCATE 0,IY:PRINT "◆";:LOCATE 28,IY:PRINT "◆";:RETURN
1330 REM ***==============RIGHT ROUTINE===============***
1340 IF IX=25 THEN RETURN ELSE IX=IX+1:COLOR 6
1350 LOCATE IX-1,0:PRINT " ◆ ";:LOCATE IX-1,24:PRINT " ◆ ";:RETURN
1360 REM ***===============LEFT ROUTINE===============***
1370 IF IX=3 THEN RETURN ELSE IX=IX-1:COLOR 6
1380 LOCATE IX-1,0:PRINT " ◆ ";:LOCATE IX-1,24:PRINT " ◆ ";:RETURN
1390 REM ***==========TITLE DISPLAY ROUTINE==========***
1400 COLOR 6
1410 LOCATE 3,4 :PRINT "░░░░ ░░░░ ░░░░ ░░░░ ░░░░";
1420 LOCATE 3,5 :PRINT "░    ░ ░ ░ ░ ░    ░   ";
1430 LOCATE 3,6 :PRINT "░    ░ ░ ░ ░ ░    ░   ";
1440 LOCATE 3,7 :PRINT "░░░░ ░░░░ ░░░░ ░    ░░░░";
1450 LOCATE 3,8 :PRINT "   ░ ░    ░ ░ ░    ░   ";
```

```
1460 LOCATE 3,9 :PRINT "   ▒ ▒    ▒  ▒ ▒    ▒   ";
1470 LOCATE 3,10:PRINT "▒▒▒▒ ▒    ▒  ▒ ▒▒▒▒ ▒▒▒▒";
1480 LOCATE 3,14:PRINT "▒▒▒▒ ▒  ▒ ▒▒▒▒ ▒▒▒▒ ▒▒▒▒";
1490 LOCATE 3,15:PRINT "▒    ▒  ▒ ▒  ▒ ▒    ▒   ";
1500 LOCATE 3,16:PRINT "▒    ▒  ▒ ▒  ▒ ▒    ▒   ";
1510 LOCATE 3,17:PRINT "▒    ▒▒▒▒ ▒▒▒▒ ▒▒▒▒ ▒▒▒▒";
1520 LOCATE 3,18:PRINT "▒    ▒  ▒ ▒  ▒    ▒▒   ";
1530 LOCATE 3,19:PRINT "▒    ▒  ▒ ▒  ▒    ▒▒   ";
1540 LOCATE 3,20:PRINT "▒▒▒▒ ▒  ▒ ▒  ▒ ▒▒▒▒ ▒▒▒▒";
1550 FOR I=0 TO 5000:NEXT:RETURN
1560 REM ***=============DATA DEFINITION==============***
1570 DATA "╲                    ╱  "                    :'N0
1580 DATA " ╲                  ╱   "
1590 DATA " │╲                ╱│   "
1600 DATA " │ ╲              ╱ │   "
1610 DATA " │  ╲            ╱  │   "
1620 DATA " │  │╲          ╱│  │   "
1630 DATA " │  │ ╲        ╱ │  │   "
1640 DATA " │  │ │╲      ╱│ │  │   "
1650 DATA " │  │ │ ╲    ╱ │ │  │   "
1660 DATA " │  │ │ │╲  ╱│ │ │  │   "
1670 DATA " │  │ │ ││└┘││ │ │  │   "
1680 DATA " │  │ │ │└  ┘│ │ │  │   "
1690 DATA " │  │ │ └────┘ │ │  │   "
1700 DATA " │  │ │╱──────╲│ │  │   "
1710 DATA " │  │ └────────┘ │  │   "
1720 DATA " │  │╱──────────╲│  │   "
1730 DATA " │  └────────────┘  │   "
1740 DATA " │ ╱──────────────╲ │   "
1750 DATA " │╱                ╲│   "
1760 DATA " └                  ┘   "
1770 DATA "╱────────────────────╲  "
1780 DATA "╲                    ╱  "                    :'N1
1790 DATA " ╲                  ╱   "
1800 DATA " │╲                ╱│   "
1810 DATA " │ ╲              ╱ │   "
1820 DATA " │  ╲            ╱  │   "
1830 DATA " │  │╲          ╱│  │   "
1840 DATA " │  │ ╲        ╱ │  │   "
1850 DATA " │  │ │╲      ╱│ │  │   "
1860 DATA " │  │ │ ╲    ╱ │ │  │   "
1870 DATA " │  │ │ │╲  ╱│ │ │  │   "
1880 DATA " │  │ │ ││└┘││ │ │  │   "
1890 DATA " │  │ │ │└  ┘│ │ │  │   "
1900 DATA " │  │ │ └────┘ │ │  │   "
1910 DATA " │  │ │╱──────╲│ │  │   "
1920 DATA " │  │ └────────┘ │  │   "
1930 DATA " │  │╱──────────╲│  │   "
1940 DATA " │  └────────────┘  │   "
1950 DATA " │ ╱──────────────╲ │   "
1960 DATA " │╱                ╲│   "
1970 DATA " └                  ┘   "
1980 DATA "╱────────────────────╲  "
1990 DATA "╲                    ╱  "                    :'N2
2000 DATA " ╲                  ╱   "
2010 DATA " │╲                ╱│   "
2020 DATA " │ ╲              ╱ │   "
2030 DATA " │  ╲            ╱  │   "
2040 DATA " │  │╲          ╱│  │   "
2050 DATA " │  │ ╲        ╱ │  │   "
2060 DATA " │  │ │╲      ╱│ │  │   "
2070 DATA " │  │ │ ╲    ╱ │ │  │   "
2080 DATA " │  │ │ │╲  ╱│ │ │  │   "
2090 DATA " │  │ │ ││└┘││ │ │  │   "
2100 DATA " │  │ │ │└  ┘│ │ │  │   "
2110 DATA " │  │ │ └────┘ │ │  │   "
2120 DATA " │  │ │╱──────╲│ │  │   "
2130 DATA " │  │ └────────┘ │  │   "
2140 DATA " │  │╱──────────╲│  │   "
2150 DATA " │  └────────────┘  │   "
2160 DATA " │ ╱──────────────╲ │   "
2170 DATA " │╱                ╲│   "
2180 DATA " └                  ┘   "
2190 DATA "╱────────────────────╲  "
```



PC-8001

スター・ウォーズ 帝国の逆襲ゲーム

映画スター・ウォーズ『帝国の逆襲』が公開になりました.氷の惑星『ホス』で反乱軍を発見した帝国軍は，四つ脚で歩く巨大な戦軍(？)—AT-ATスノー・ウォーカー（全地域用装甲歩行機）—で，反乱軍の基地に攻めてきます．この戦闘シーンは，見せ場のひとつになっていますが，今回これをヒントにしたゲームを作ってみました.

京都薬科大学マイクロコンピュータ同好会
馬嶋　量

ゲームの説明

画面の右側から８機のスノー・ウォーカーが，レーザーを発射しながらゆっくり前進してきます．

画面の左半分に４列のバリケードがあり，左端が反乱軍の基地です．最初に２人の兵士が基地内に待機していて，外に出ている兵士が戦います．

この兵士はテンキーの[8]で画面上方へ，[2]で下方へ移動でき，同時に[5]を押すと，より大きく動くことができます．

また[6]で前方，[4]で後方のバリケードを乗り越えるために姿勢を変えていきます(図１)．

兵士は先頭のスノー・ウォーカーの頭の位置よりも右側にいることはできません．兵士の位置までスノー・ウォーカーが侵攻してくると，兵士は自動的に１列後ろに移動します．

上下方向のキー[8]，[2]（[5]）と前後方向のキー[6]，[4]を同時に使って斜めに移動することもできます．

兵士は銃をかまえているときだけ，テン・キーの[0]を押せば水平方向に発砲します．弾丸はバリケードを貫通する（その部分は崩れる）ので，バリケードの後ろからの攻撃も可能です．

スノー・ウォーカーは水平方向±7.4度の範囲でレーザーを発射します．レーザー攻撃はバリケードを貫通することはできませんが，崩れかけた部分からは突き抜けてきます．ただし，基地内には届きません．

スノー・ウォーカーは2,000ユニットのエネルギーを持っていて，兵士の弾丸を胴体に受けるとそのエネルギーは減少します．兵士が最前列のバリケードの前に出て発射したときには命中したスノーウォーカーのエネルギーが1,700ユニット減少し，兵士がより後方から発射するに従って，それぞれ，850，550，350，250ユニット減少します．

また，スノー・ウォーカーは１歩前進したときに，そのエネルギー量に応じた色に変化します((白黒では暗くなっていきます)．そして，200ユニット以下になると前進しなくなり，エネルギーがなくなると前脚から崩れて機能を停止します(図２，３)．

兵士は，スノー・ウォーカーのレーザーに命中すると死んでしまいます(図４)．代わって，待機していた兵士の１人が基地から出てきます(再び基地内には入れません)．

８機のスノー・ウォーカーを倒すと画面が新しくなり，新たに８機がより近くから攻撃してきます．

得点は，兵士が発砲するごとに－180点，スノー・ウォーカーを倒すごとに＋2,000点が加算され,スノー・ウォーカーを倒したときに表示が新しくなります．

最初の８機を倒して画面が新しくなったときに，兵士が１人増えます．すべての兵士が死ぬか，スノー・ウォーカーが基地の直前に達するとゲームは終了します．そして，そのゲームの得点と今までの最高得点が表示されます．

[n]以外のキーを押すと，再びゲームがスタートします．

帝国の逆襲ゲーム

写真1　ゲームが開始され，STAR WARSの画面が出てくる．



写真3　スノー・ウォーカーに撃たれ，反乱軍の兵士が死んだところ．

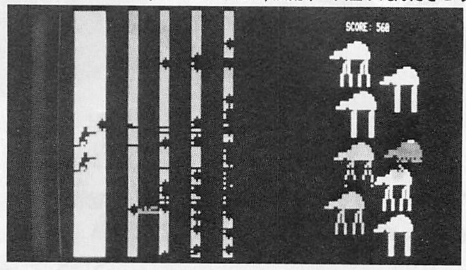


写真2　スノー・ウォーカーが反乱軍の兵士を目がけて撃ってきた．

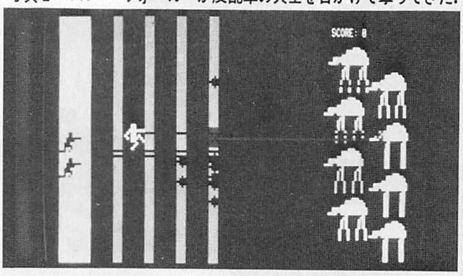


写真4　スノー・ウォーカーが死んでいくところ．

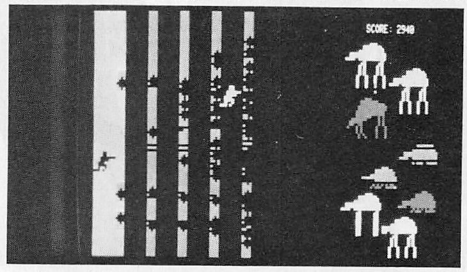


図1　前後移動

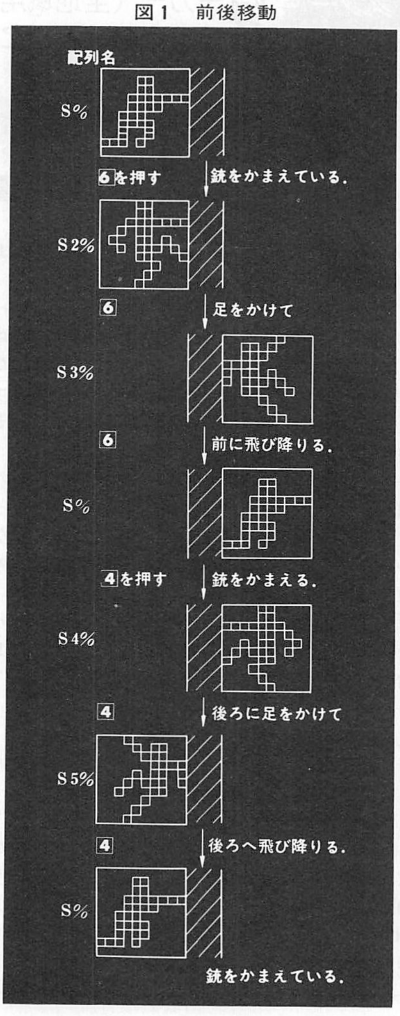


図2　スノー・ウォーカーの歩行

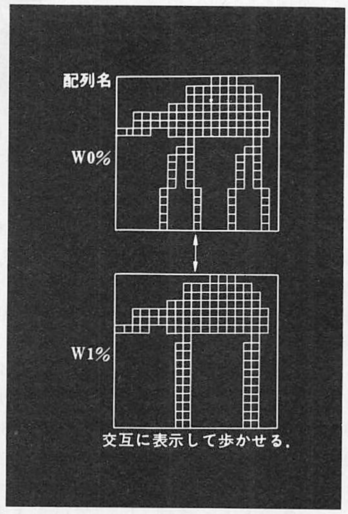


図3　スノー・ウォーカーの被弾

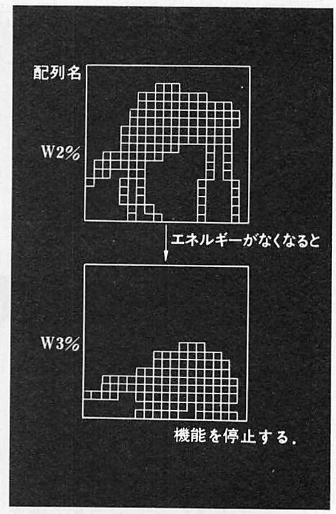
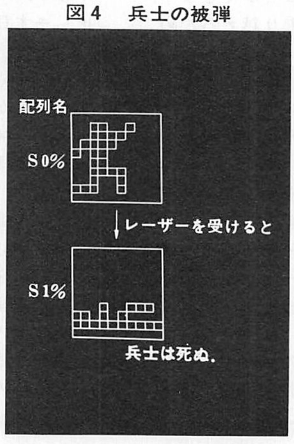


図4　兵士の被弾



プログラムの説明

このプログラムはPC-8001のBASICで書いてあります．16K RAMでも使えるように，また高速化するためにマルチ・ステートメントを多用してできるだけ短かくしたので，たいへん見にくくなってしまいました．

これで，約9Kのメモリを使います．このプログラムは各ブロックごとにバラバラに作ったので，さらに整理すればもう少し短かくすることが可能です．

表1　変数表

a）配列変数　PUT@のデータ

変数名	内容
Z%(124)	タイトル
W0%(22)	スノー・ウォーカー歩行1
W1%(22)	〃　〃2
W2%(22)	〃　分解1
W3%(22)	〃　〃2
S%(6)	兵士　銃かまえ
S0%(6)	〃　被弾1
S1%(6)	〃　〃2
S2%(6)	〃　前進1
S3%(6)	〃　〃2
S4%(6)	〃　後退1
S5%(6)	〃　〃2
B1%(1)	爆発1
B2%(2)	〃　2

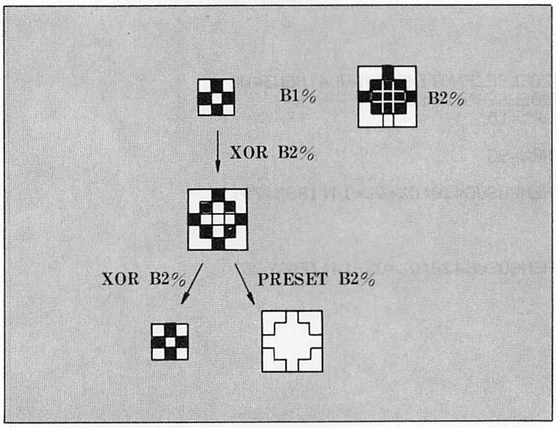
b）その他の配列変数

変数名	内容
U%(7)	スノー・ウォーカーX座標
V%(7)	〃　Y座標
E%(7)	〃　エネルギー量
X%(10, 7)	レーザー軌跡X座標
Y%(10, 7)	〃　Y座標
A%(4)	被弾して失うエネルギー量

c）数値変数

変数名	内容
Q%	兵士のX座標
R%	〃　Y座標
P%	〃　X座標の制限値
M%	〃　姿勢
G%	〃　発砲による失点
C%	〃　残り数
QC%	待機兵士のX座標
RC%	〃　Y座標
C0%	〃　消去位置
WB%	命中したスノー・ウォーカーの番号
UB%	命中点のX座標
WC%	〃　高さ
F%	被弾して失うエネルギー量
Z%	スノー・ウォーカー全滅の判定
H	得点
HS	最高得点
MA%	スノー・ウォーカーの出現位置の指定
N%	〃　番号
X%	〃　レーザー発射X座標
Y%	〃　〃　Y座標
T	〃　〃　角度
PO%	〃　〃　命中判定
L%	〃　〃　命中点の指定
W	タイルのわくの大きさを指定
ME%	画面の数
D	タイマ
I%	ループ・カウンタ
J%	〃
K%	〃

図5　命中点での爆発パターンの変化



高速化のためほとんどを整数変数として，『%』を付けて使っていますが，これはベンチマーク・テストの結果が**DEFINT**で宣言するよりも速かったためです．しかし，その差はわずかで，宣言する方が“%”の分だけメモリが少なくてすむので良いかもしれません．

プログラムの構造が簡単なので，流れ図は省略して，**リスト1**に従って説明します．**表1**に主な変数の一覧表を示します．

まず，最初に0行がいくつも並んでいますが，これは他の番号でプログラムを入力してからモニタの**S**コマンドで行番号の入っている番地の内容を0に変更したものです．1行も特に意味はないので，100行から入力してください．

100～190行は，まず910～980行のサブルーチンでタイトルと説明を表示して，タイトルで使った配列Z%をERASEしてからDATAを各配列変数に読み込んでいます．

200～280行では，画面設定を行ないます．

300～470行は，反乱軍兵士の移動に関するところです．兵士の姿勢に応じて430～470で，それぞれの姿勢の兵士を表示します．

500～540行はスノー・ウォーカーの移動で，1歩進むごとに2つのパターンを交互に表示して歩かせています（図2）．

600～665行は兵士の攻撃でスノー・ウォーカーの胴体に命中すると，そのスノー・ウォーカーのエネルギーを減らします．

N-BASICでは，色をキャラクタ1文字分のエリアごとに指定するので，グラフィックで書かれたところに他の色で重ねて書くと，1文字分のエリア内のドットがすべてあとから書いた色に変わってしまいます．それで，PRESETのときに元の色を指定して色を戻しています．

700～770行はスノー・ウォーカーの攻撃です．何かに命中したことを判定するために，280行のデータで指定したX座標ごとにレーザーの軌跡のY座標を計算して，その点のドットがセットされているかどうか**POINT（X，Y）**で調べています．そして，命中したところまでレーザーの軌跡を表示します．この部分は工夫すれば，さらに高速化できるでしょう．

800～840行ではゲームが終了したときに得点を計算して表示します．再ゲームのときは，180行へジャンプします．

おわりに

このゲームで兵士が攻撃する位置に対するスノー・ウォーカーが失うエネルギー量を決めるにあたって，何人もの人にテストしてもらって経験的に最適値を求めました．

また，コムスポット共立，および，京都薬科大学マイクロコンピュータ同好会のみなさんにいろいろと協力いただきました．たいへんありがとうございました．

帝国の逆襲ゲーム　プログラム・リスト

```
_0 '*********************************
0 '***                           ***
0 '***         STAR  WARS        ***
0 '***         the ENPIRE        ***
0 '***        strikes back       ***
0 '***             by            ***
0 '***       USO lab.  1980      ***
0 '***                           ***
0 '*********************************
1 '#00000
100 'ｼｮｷｾｯﾃｲ 1
110 WIDTH80,25:CONSOLE0,25,0,1:COLOR7,0,1
120 GOSUB910
125 DEFDBL H
130 DIMU%(7),V%(7),E%(7),X%(10,7),Y%(10,7),A%(4)
140 DIMW0%(22),W1%(22),S%(7),S2%(7),S3%(7),S4%(7),S5%(7),B1%(1),B2%(2),S0%(7),S1%
(7),W2%(22),W3%(22)
145 RESTORE190:FORI%=0TO4:READA%(I%):NEXT
150 RESTORE1010:FORI%=0TO22:READW0%(I%),W1%(I%),W2%(I%),W3%(I%):NEXT
160 FORI%=0TO7:READS%(I%),S0%(I%),S1%(I%),S2%(I%),S3%(I%),S4%(I%),S5%(I%):NEXT
170 READB1%(0),B1%(1),B2%(0),B2%(1),B2%(2)
180 X%=0:Y%=0:T=0:I%=0:J%=0:Q%=23:R%=53:F%=0:G%=0:M%=3:C%=3:ME%=0:N%=0:H=0
190 DATA25,35,55,85,170
200 'ｼｮｷｾｯﾃｲ 2
210 IFME%>4THENMA%=4ELSEMA%=ME%
220 FORI%=0TO7:E%(I%)=200:U%(I%)=119-(MA%-I%MOD2)*20:V%(I%)=18+I%*10:X%(0,I%)=U%
(I%):Y%(0,I%)=V%(I%):RESTORE280:FORK%=1TO10:READX%(K%,I%):IFX%(K%,I%)>=U%(I%)THE
NX%(K%,I%)=U%(I%)-1
225 NEXTK%,I%:P%=U%(0)-1
230 PRINTCHR$(12);:WIDTH80,25
240 LINE(13,99)-(0,0),PSET,5,BF:FORI%=24TO66STEP14:LINE(I%+3,99)-(I%,0),PSET,6,B
F:NEXT
250 COLOR7:FORI%=0TO7:IFU%(I%)MOD2=1THENPUT@(U%(I%),V%(I%)-6)-(U%(I%)+18,V%(I%)+
11),W0%,PSET:NEXTELSEPUT@(U%(I%),V%(I%)-6)-(U%(I%)+18,V%(I%)+11),W1%,PSET:NEXT
260 IFC%=1THEN410ELSECOLOR5:QC%=0:FORRC%=55TO75-10*C%STEP-10:PUT@(QC%,RC%)-(QC%+
9,RC%+9),S%,PRESET:NEXT
270 COLOR7:LOCATE60,1:PRINT"SCORE:";H*10;:GOTO410
280 DATA 75,68,61,54,47,40,33,26,19,12
300 'ﾍｲｼ ｲﾄﾞｳ
310 IFQ%>P%THENM%=2:GOTO390ELSEIFNOTINP(1)AND1THEN320ELSEIFNOTINP(0)AND4THEN340E
LSEIFNOTINP(0)AND64THEN360ELSEIFNOTINP(0)AND16THEN390ELSEGOTO510
320 GOSUB420:R%=R%-2+((INP(0)AND32)=0)*5:IFR%<15THENR%=15
330 GOTO350
340 GOSUB420:R%=R%+2-((INP(0)AND32)=0)*5:IFR%>93THENR%=93
350 IFNOTINP(0)AND16THEN390ELSEIFINP(0)AND64THEN410
360 M%=M%+1:IFM%>5THENM%=3:GOTO410ELSEIFM%=2ORM%=5THENGOSUB420:Q%=Q%+14:IFQ%>79T
HENQ%=79
370 IFM%=2THENM%=5
380 GOTO410
390 M%=M%-1:IFM%<1THENM%=3:GOTO410ELSEIFM%=1ORM%=4THENGOSUB420:Q%=Q%-14:IFQ%<23T
HENQ%=23
400 IFM%=4THENM%=1
410 COLOR7:ONM%GOTO430,440,470,460,450
420 LINE(Q%-9,R%-3)-(Q%,R%+6),PRESET,,BF:RETURN
430 PUT@(Q%-9,R%-3)-(Q%,R%+6),S5%,PSET:GOTO510
440 PUT@(Q%-9,R%-3)-(Q%,R%+6),S4%,PSET:GOTO510
450 PUT@(Q%-9,R%-3)-(Q%,R%+6),S3%,PSET:GOTO510
460 PUT@(Q%-9,R%-3)-(Q%,R%+6),S2%,PSET:GOTO510
470 PUT@(Q%-9,R%-3)-(Q%,R%+6),S%,PSET:GOTO510
500 'ｽﾉｰｳｫｰｶｰ ｲﾄﾞｳ
510 N%=8*RND(1):IFV%(N%)=0THENIFRND(1)<.5THEN610ELSEGOTO510
520 IFE%(N%)>20THENU%(N%)=U%(N%)-1:FORI%=0TO10:IFX%(I%,N%)>=U%(N%)THENX%(I%,N%)=
U%(N%)-1:NEXTELSENEXT
525 IFU%(N%)=P%THENP%=U%(N%)-1
530 COLORE%(N%)¥30+1:IFU%(N%)MOD2THENPUT@(U%(N%),V%(N%)-6)-(U%(N%)+18,V%(N%)+11)
,W0%,PSETELSEPUT@(U%(N%),V%(N%)-6)-(U%(N%)+18,V%(N%)+11),W1%,PSET
540 IFU%(N%)<24THENLINE(0,0)-(11,25),"X",2,BF:FORD=0TO1500:NEXT:GOTO810ELSEGOTO6
10
600 'ﾍｲｼ ｺｳｹﾞｷ
610 IFNOTINP(0)AND1ANDM%=3THENWB%=(R%-12)¥10+(R%>91):WC%=(R%-12)MOD10:F%=A%((Q%-
23)¥14)ELSEGOTO710
620 IFV%(WB%)=0ORWC%>6ORR%>91THENBEEP1:LINE(156,R%)-(Q%,R%),PSET,5:LINE(156,R%)-
(Q%,R%),PRESET,6:BEEP0:GOTO650
630 UB%=U%(WB%)-WC%*2+12:COLOR5:BEEP1:LINE(UB%,R%)-(Q%,R%),PSET:PUT@(UB%-1,R%-1)
-(UB%+1,R%+1),B1%,PSET:COLOR6:LINE(UB%,R%)-(Q%,R%),PRESET:PUT@(UB%-2,R%-2)-(UB%+
2,R%+2),B2%,XOR:E%(WB%)=E%(WB%)-F%:PUT@(UB%-2,R%-2)-(UB%+2,R%+2),B2%,XOR:BEEP0
640 IFE%(WB%)<1THENCOLOR1:PUT@(U%(WB%),V%(WB%)-6)-(U%(WB%)+18,V%(WB%)+11),W2%,PS
ET:BEEP:FORD%=0TO20:NEXT:PUT@(U%(WB%),V%(WB%)-6)-(U%(WB%)+18,V%(WB%)+11),W3%,PSE
T:V%(WB%)=0:Z%=0:GOTO660
650 G%=G%+18:GOTO710
660 P%=138:BEEP1:FORI%=0TO7:IFV%(I%)>0ANDU%(I%)<=P%THENP%=U%(I%)-1
665 Z%=Z%+V%(I%):NEXT:BEEP0:G%=G%+18:H=(200-G%)+H:G%=0:LOCATE60,1:COLOR7:PRINT"S
CORE:";H*10;:IFZ%THEN710ELSEME%=ME%+1:C%=C%-(ME%=1):GOTO210
700 'ｽﾉｰｳｫｰｶｰ ｺｳｹﾞｷ
710 IFRND(1)>.5THENN%=(R%-12)¥10+(R%>91)ELSEN%=8*RND(1)
720 IFV%(N%)=0THENIFRND(1)>.5THEN710ELSEFORD%=0TO30:NEXT:GOTO310ELSEY%=Y%(0,N%):
X%=X%(0,N%):T=RND(1)*.26-.13:B%=0:COLOR2
730 FORI%=1TO10:Y%(I%,N%)=Y%-(X%-X%(I%,N%))*TAN(T):PO%=POINT(X%(I%,N%),Y%(I%,N%)
):IFPO%ORI%=10THENL%=I%:I%=10:NEXTELSENEXT
```

帝国の逆襲ゲーム　プログラム・リスト

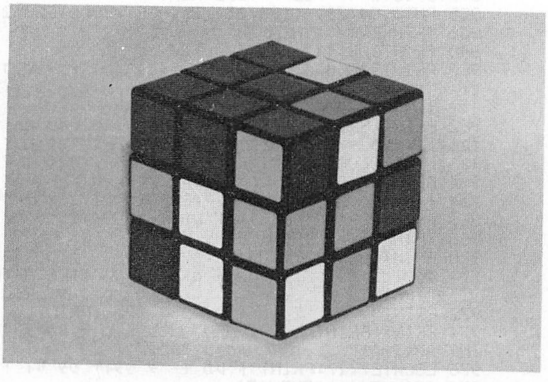
```
735 BEEP1:BEEP0:FORI%=1TOL%:LINE(X%(I%,N%),Y%(I%,N%))-(X%(I%-1,N%),Y%(I%-1,N%)),
PSET:NEXT
740 PUT@(X%(L%,N%)-1,Y%(L%,N%)-1)-(X%(L%,N%)+1,Y%(L%,N%)+1),B1%,PSET:FORJ%=1TOL%
:LINE(X%(J%,N%),Y%(J%,N%))-(X%(J%-1,N%),Y%(J%-1,N%)),PRESET,6:NEXT
750 BEEP1:PUT@(X%(L%,N%)-2,Y%(L%,N%)-2)-(X%(L%,N%)+2,Y%(L%,N%)+2),B2%,XOR:COLOR6
:PUT@(X%(L%,N%)-2,Y%(L%,N%)-2)-(X%(L%,N%)+2,Y%(L%,N%)+2),B2%,PRESET:BEEP0:IFX%(L
%,N%)<>Q%-4THEN310
760 COLOR2:PUT@(Q%-9,R%-3)-(Q%,R%+6),S0%,PSET:BEEP1:FORD%=0TO300:NEXT:COLOR1:PUT
@(Q%-9,R%-3)-(Q%,R%+6),S1%,PSET:FORD%=0TO300:NEXT:BEEP0:PUT@(Q%-9,R%-3)-(Q%,R%+6
),S1%,PRESET:C%=C%-1
770 IFC%=0THEN810ELSEC0%=65-C%*10:LINE(0,C0%)-(9,C0%+9),PSET,5,BF:M%=3:Q%=23:R%=
53:GOTO410
800 'ﾄｸﾃﾝ ﾋｮｳｼﾞ
810 H=H-G%
815 IFH>HSTHENHS=H
820 WIDTH40,25:PRINTCHR$(12):COLOR7:LOCATE7,6:PRINT"ｻｲｺｳ ﾄｸﾃﾝ ﾊ  ";HS*10;" ﾃﾞｽ":
LOCATE7,9:PRINT"ｱﾅﾀ ﾉ ﾄｸﾃﾝ ﾊ ";H*10;" ﾃﾞｽ":FORD=0TO1500:NEXT:LOCATE7,15:PRINT"ﾓｳ
 ｲﾁﾄﾞ ﾁｮｳｾﾝ ｼﾏｽｶ ? (y/n)"
830 LOCATE29,17:IFINPUT$(1)<>"n"THENPRINT"Yes";GOTO180ELSEPRINT:PRINTTAB(7);"Goo
d-by !"
840 END
900 'ｾﾂﾒｲ
910 DIMZ%(124):RESTORE2010
920 FORI%=0TO124:READZ%(I%):NEXT:PRINTCHR$(12):FORW=1TO5:LINE(54-W*8,28-W*4)-(10
8+W*8,63+W*4),PSET,W+2,B:NEXT
925 PUT@(54,28)-(108,63),Z%,PSET:LOCATE28,23:PRINT"ﾄﾞﾚｶ ｷｰ ｦ ﾀﾀｲﾃ ｸﾀﾞｻｲ !"
930 IFINKEY$=""THEN930
940 PRINTCHR$(12):WIDTH40,20
960 PRINTTAB(4);"ｽﾀｰ･ｳｫｰｽﾞ ﾃｲｺｸ ﾉ ｷﾞｬｸｼｭｳ ｹﾞｰﾑ":PRINT:PRINT:PRINT" ｺｺ ﾊ ｺｵﾘ ﾉ ﾜｸ
ｾｲ < ﾎｽ > ﾃﾞｽ.":PRINT:PRINT" ｱﾅﾀ ﾊ ｾﾏﾘｸﾙ ﾃｲｺｸ･ｸﾞﾝ ﾉ AT-AT･ｽﾉｰｳｫｰｶｰ ｶﾗ ﾐｶﾀ ﾉ ｷﾁ ｦ
 ﾏﾓｯﾃ ｸﾀﾞｻｲ."
962 PRINT:PRINT" ｽﾉｰｳｫｰｶｰ ﾊ ﾊｼﾞﾒ ﾆ 2000 unit ﾉ ｴﾈﾙｷﾞｰ ｦﾓｯﾃ ｲﾏｽ. ｽﾉｰｳｫｰｶｰ ﾉ ｲﾛ ﾊ
ｴﾈﾙｷﾞｰ ﾘｮｳ ｦ ｱﾗﾜｼ, ｼﾛ ";:COLOR6:PRINT"ｷｲﾛ ";:COLOR5:PRINT"ﾐｽﾞｲﾛ ";:COLOR4:PRINT"
ﾐﾄﾞﾘ ";:COLOR3:PRINT"ﾑﾗｻｷ ";:COLOR2:PRINT"ｱｶ ";:COLOR1:PRINT"ｱｵ ";:COLOR7
964 PRINT"ﾉ ｼﾞｭﾝ ﾆ ｽｸﾅｸ ﾅﾘﾏｽ.":PRINT:PRINT" ﾄﾞｳﾀｲ ﾆ ﾒｲﾁｭｳ ｽﾙ ﾄ 250-1700 unit ﾉ ｴ
ﾈﾙｷﾞｰ ｦ ｳﾊﾞｳｺﾄ ｶﾞ ﾃﾞｷﾏｽ.":PRINT" ﾏｴ ﾆ ﾃﾞﾃ ｳﾂ ﾎﾄﾞ ｷｮｳﾘｮｸ ﾃﾞｽ."
966 PRINT:PRINTTAB(7);"ﾄﾞﾚｶ ｷｰ ｦ ﾀﾀｲﾃ ｸﾀﾞｻｲ !"
968 IFINKEY$=""THEN968
970 PRINTCHR$(12);TAB(4);"ｽﾀｰ･ｳｫｰｽﾞ ﾃｲｺｸ ﾉ ｷﾞｬｸｼｭｳ ｹﾞｰﾑ":PRINT:PRINT:PRINT" 3 ﾆﾝ
 ｶﾞ ﾀﾀｶｲﾏｽ ｶﾞ, 8 ｷ ﾀｵｽ ﾄ ﾋﾄﾘ ﾌｴﾏｽ.";:PRINT:PRINT"      8          5 ｦ ｵｼﾅｶﾞﾗ 8 ﾏﾀﾊ
 2 ｦ":PRINT"    4+6        ｵｽ ﾄ, ﾊﾎｸ ｳｴ ﾏﾀﾊ ｼﾀ ﾆ"
972 PRINT"    2           ｳｺﾞｷ ﾏｽ｡":PRINT"      0･･･ﾊｯｼｬ"
974 PRINT:PRINT"    1 ﾊﾟｯ ｳﾂ ｺﾞﾄ ﾆ -180 ﾃﾝ":PRINT"    1 ｷ ﾀｵｽ ｺﾞﾄ ﾆ +2000 ﾃﾝ":PR
INTTAB(20);"... ｶﾞ,  ﾄｸﾃﾝ ｻﾚﾏｽ."
976 LOCATE7,17:PRINT"ﾄﾞﾚｶ ｷｰ ｦ ﾀﾀｲﾃ ｸﾀﾞｻｲ !":PRINT
978 IFINKEY$=""THEN978
980 PRINTCHR$(12):LOCATE5,10:PRINT"MAY THE FORCE BE WITH YOU":ERASEZ%:RETURN
999 'DATA ******
1000 REM W0,W1,W2,W3,(0-22) ***
1010 DATA 342,342,342,342,14336,14336,0,0
1020 DATA -2048,-2048,0,0,-16377,-16377,-16384,0
1030 DATA -32641,-32641,-32767,0,-15361,-15361,127,0
1040 DATA 16383,16383,8188,0,-2,-2,-16,0
1050 DATA -195,-195,-127,0,8207,8207,-497,0
1060 DATA -32736,-32736,-4033,0,385,385,-32257,0
1070 DATA 2570,3084,2623,448,20560,24672,20607,16320
1080 DATA -32128,768,-32516,-512,5122,6147,1634,-1021
1090 DATA 4116,-16360,10388,-481,-32495,192,16656,-3585
1100 DATA 2184,1542,2177,-4097,17476,12336,17418,32761
1110 DATA 8736,-32384,8352,-21888,4354,3073,2306,14849
1120 DATA 17,12,17,31
1130 REM S,S0,S1,S2,S3,S4,S5,(0-7) ***
1140 DATA 100,100,100,100,100,100,100
1150 DATA -16384,12288,0,-16336,-20416,-16336,16392
1160 DATA 768,-32063,0,16256,-15936,2032,3587
1170 DATA 30974,3335,0,-19955,11523,13356,-12048
1180 DATA -16272,-16336,-32768,1488,-16012,4841,19362
1190 DATA 2307,2307,2780,4131,1032,515,2060
1200 DATA 23,39,1023,16416,32,-32752,8208
1210 DATA 0,0,0,0,1,0,0
1220 REM B1(0-1),B2(0-2) ***
1230 DATA 9,170,25,32196,71
2000 REM Z%(0-124) ***
2010 DATA 1980,0,0,0,124,0,0,8688
2020 DATA 0,0,1984,22,480,7936,3064,-27136
2030 DATA 31744,-32,-17403,-4038,-127,-15617,-2747,-8697
2040 DATA 1023,-3567,18434,4071,4224,-8119,14271,22528
2050 DATA -32624,4687,-8192,2031,10238,-16384,14323,-8
2060 DATA 3,15356,-40,63,-272,-5091,255,31200
2070 DATA -281,1014,-512,15293,15111,28672,-8449,-19569
2080 DATA 28701,29628,-12305,-4385,-8325,30653,28655,-513
2090 DATA -12305,1851,-80,-2050,7649,-9664,-241,-4037
2100 DATA -17216,-32532,7647,20480,30257,-6720,-8178,26788
2110 DATA -4165,-16272,24147,-636,119,30767,-16183,1022
2120 DATA 24741,164,16254,-13636,7745,8380,21184,15395
2130 DATA -27968,15312,-2385,21504,10375,21128,-16896,16042
2140 DATA 17508,-2046,24528,16896,-8190,30723,9,-32707
2150 DATA 15,168,128,62,1024,16384,248,0
2160 DATA 0,992,0,0,0
```



RUBIK CUBE
ルービック・キューブの解き方
立方体パズルを解剖する

神谷　慎吾

はじめに



数カ月程前から，百貨店などで写真のようなパズルが売り出されています．このパズル，発売当初は売り場に山のように積んであったという話ですが，新聞などでその存在が紹介されて以来売り上げが急に伸び，一頃はあまりの売れ行きに品切れの店が続出した，といううわさもあるほどです．そのようなわけですから，ご覧になった方や実際に買われた方もかなり多いのではないかと思われます．

もともとこのパズルはハンガリーのあるデザイナーによって発明されたものだそうで，欧州のパズル愛好家の間では既に良く知られていたようです．そのパズルが遂に日本にも上陸，といういきさつなのですが，上の状況から見ますに，日本でもこのパズルが静かなブームを呼び起しつつあるのはまず間違いないでしょう．

ところで，このパズルを手に取って遊ばれた方ならご存知かと思いますが，話に聞くよりやってみると以外なむずかしさで，試行錯誤にたよっていたのではいつまでたってもらちがあきません．この辺の事情が，かつて流行したペントミノというはめ込みパズル（プラパズルNo.5という商品名で現在でも市販されています）とは大きく異なっています．そこで，少しこの2つを対比してみましょう．

まずペントミノですが，このパズルを解くには本質的に試行錯誤を行なうしかないようです．にもかかわらず，人間が試しても案外解が見つかりますし，計算機を使って同様のことを行なえば全解を得るのも容易です．その意味で，ペントミノは簡単なパズルであるということができるかも知れません．

一方，この立方体パズルはどうかを考えてみます．試行錯誤にたよってはまず解く（＝色を揃える）ことができません．それなら計算機にやらせればよいと考えたくなりますが，残念ながら現在の計算機の記憶容量では解の探索にまったく不充分です．

本来，ペントミノのプログラムに使われている深さ優先の探索と，立方体パズルの解を見い出すための広さ優先の探索とでは探索の方式がかなり違います．広さ優先の探索では一般に非常に多くの記憶場所を必要とするので，扱えるパズルの複雑さが制限されてきます．詳しい話は文献1をご覧ください．

現在この後者の探索法で解が求まっているパズルとしては，箱入り娘（MOON-LANDERという商品名のものもありますが，最近どちらもあまりみかけません）などがあります．

こう書くと打つ手がないかのように聞えますがそうでもありません．実は，『群論』という数学の助けを借りると，このパズルの状態の変化を非常に見通し良く解析することができるのです．この接近法により，具体的なパズルの解法（＝色を揃えるための方針）も求まっています．

そこで，読者諸氏の中にもこのパズルに興味をお持ちの方は多いと思うので（工学社編集室にも何個か置いてあります），I/Oの記事としては異例ですが，上述の方法により求められた立方体パズルの一つの解法をここにご紹介したいと思います．

ですから，表題の『立方体パズルを解剖する』というのは，正確には『立方体パズルの解法を解剖する』という意味なのです．パズルをトンカチでたたいて分解し，内部の巧妙かつシンプルな機構を解明しようという意味では決してありません．

しかしながら，I/Oは数学雑誌ではありません（はずです）から，パズルのふるまいをいかに記述し，どのように解析するかに関する群論の議論は一切省き，結果として得られる解法の手続きのみお話しすることにします．この部分だけでも充分長くなります．



2 パズルの形状と遊び方

パズルの形状は 写真からも明らかなように，立方体（サイコロの形）です．さらによくみると，この全体としての立方体（以後，puzzle）が，図1のように27個の小さな立方体（以後，cube）から成り立っていることがわかります．

このpuzzleの各面が異なる6色で塗り分けられており，それぞれ図2のとおりその面内で左右に自由に回転できるようになっています（ただし，パズルの構造上，左（＝反時計回り）にばかり回すのはよくないそうです）．

面は6面あり，その上各面の回転には，右（＝時計回り）に90度の回転，左に90度の回転，180度の回転（これはどちらから回しても同じ）という3種類がありますから，合計6×3＝18通りの異なる回転が可能です．これらを**基本操作**と呼び，基本操作を1回行なうごとに1手と数えることにします．

ですから，たとえば1つの面を右に90度回転する操作を続けて2回行なったとき，結果としては180度の回転と同じなるので，，この場合は2手と数えず1手と数えます．

基本操作を何回か適用すれば，色のパターンは様々に変化します．色の揃った状態（初期状態）から始めて5～6手の操作で，既にかなりランダムなパターンになります．このランダムなパターンの美しさを味わうのも1つの楽しみ方ではありますが，普通は色がランダムに入り混じると，元のように揃えたいと考えるのが人情です．

というわけでおわかりのように，このパズルを解くというのは任意のランダムなパターンから出発して，基本操作をうまく適用することにより，puzzleを元のような色の揃った状態に戻すことに他なりません．

それでは以下このパズルの解法の説明に移りますが，混乱を避けるため，初期状態における各面の色の配置は図3のようになっているとして話を進めます．この配置は工学社に置いてあるパズルのものですが，場合によってはこれ以外の色配置の製品もあるかもしれませんので，お持ちのパズルの色が図3と異なっている方は，以下の部分をお読みになる際適当に色の置換を行なってください．

また，色とは別に説明の都合上，図4のように面にA，B…などの記号を割り当てることがありますが，これは各面と各cubeとの相対的な位置関係を便宜的に示すためのものです．

そのとき，Aという記号でAを割り当てられた面そのものを指し示すだけでなく，Aの面を右に90度回転する操作をも表わすという記号の乱用を行なうのでご注意ください．同じようにして，Aの面を左に90度回転する操作はA^{-1}で，Aの面を180度回転する操作はA^2で表しますが，A以外の記号の場合もまったく同様です．

3 パズルの解法

puzzleをよく観察してみると、色の塗られているcubeは，色を塗られている面の数によって，次の3種に分類できることがわかります（図5）．

❶Ⅰ型…puzzleの各面の中央に位置していて，1面に色を塗られた合計6個のcube．

❷Ⅱ型…puzzleの各稜に位置していて，2面に色を塗られた合計12個のcube．

❸Ⅲ型…puzzleの各頂点（角）に位置していて，3面に色を塗られた合計8個のcube．

基本操作によってⅠ型は自転しますが，配色は変わりません．すなわち，初期状態の色の配置というのはこのⅠ型の色の配置のことなのです．したがって，このⅠ型の色を

図 1

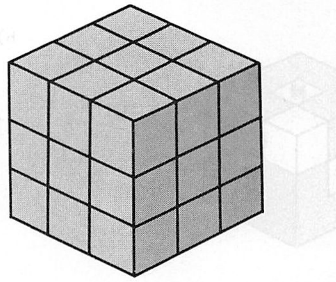
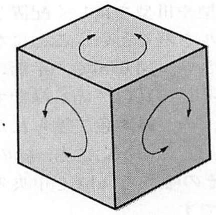
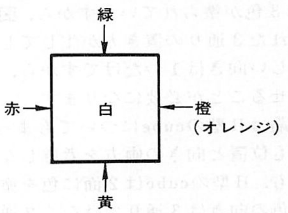


図 2



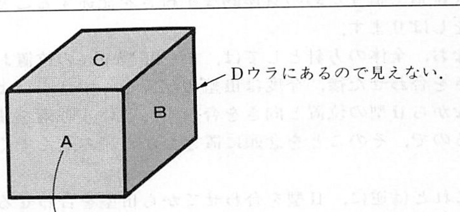
裏の面も同様

図 3



白の面を上にして置いた図．
底の面の色は青になる．

図 4



Aという記号で，この面自体を示すと同時に，
この面を右に90度回転する操作をも表す．

図 5

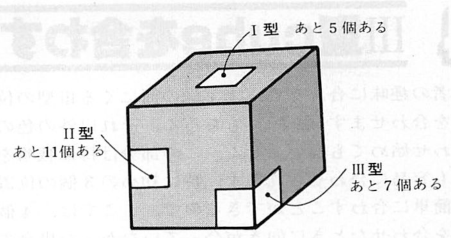
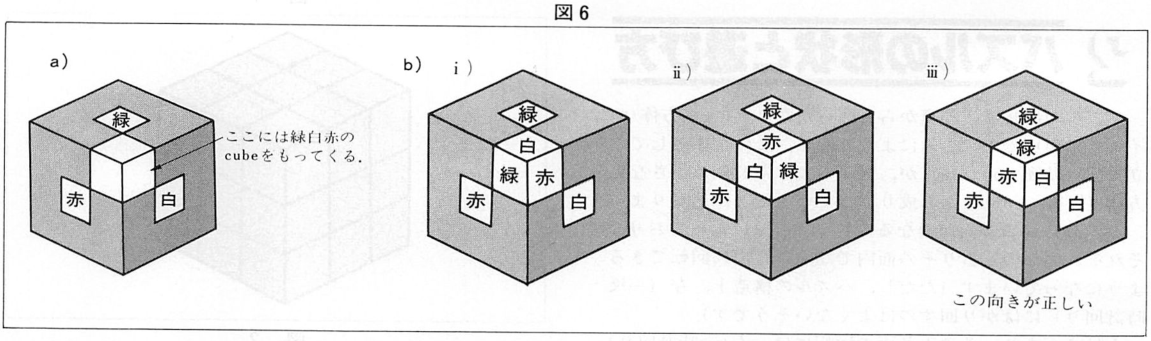


図6



基準にして他のII型やIII型を正しく配置し直す手続きを求めることが，パズルを解くということになります．

図6をご覧ください．図6(a)で，手前に見える3面の色(＝I型の色)はそれぞれ白，赤，緑ですから，図の中央に見える角には白，赤，緑の色で塗られたcubeを持ってくる必要があります．ところが白，赤，緑のcubeは1つしかありませんから，その唯一のcubeを中央の角に持ってこなければならないのです．

言い換えると，III型のcubeの正しい位置は各cubeにつき1箇所しかありません．その上，位置が合っていても，III型cubeには3色が塗られていますから，図6(b)のi)，ii)，iii)に示された3通りの置き方が生じてしまいます．これらのうち正しい向きは1つだけですから，位置の他に色の向きも合わせることが必要になります．

以上の議論はII型のcubeについてもまったく同様で，II型についても位置と向きの両方を考慮しなければなりません．もっとも，II型のcubeは2面に色を塗られているのみですから，色の向きは3通りではなく2通りになります．

ここまでの話で，cubeを正しく配置し直すという言葉の意味がおわかりいただけたと思うので，これから先は，cubeを配置し直すための具体的な手続きを記述することに話題をしぼります．

なお，全体の方針としては，まずIII型cubeの位置および向きを合わせた後，今度はIII型の配置をくずさないようにしながらII型の位置と向きを合わす，という戦術を使っているので，そのことを念頭に置きながら読み進んでください．

これとは逆に，II型を合わせてからIII型を合わせる，という方式も考えられますが，どうも能率が良くないようです．puzzleをお持ちの方は実際に操作をしながら読めば最高です．

4 III型cubeを合わす

筆者の趣味に合わせて，まず緑の面にくるIII型の位置と向きを合わせます(図7)．もちろん，それ以外の色の面から合わせ始めても構いません．この部分は特に戦略を使わずとも容易に合わせられます．特に初めの3個の位置と向きは簡単に合わすことができるので，ここでは，4個目の位置を合わせたときに向きが合っていなかった場合の対応法のみ示します．

図8をご覧ください．もしも，4個目のcubeの向きが，図8(a)のようにずれてしまった場合は，まずB^{-1}(Bの面を左90度回転)を行ないます．続いて，C，B，C^{-1}，B^{-1}，C，Bを行なえば，7手で向きが合います．

図8(b)のようにずれたときは，(a)の場合とは対称的に，

図7

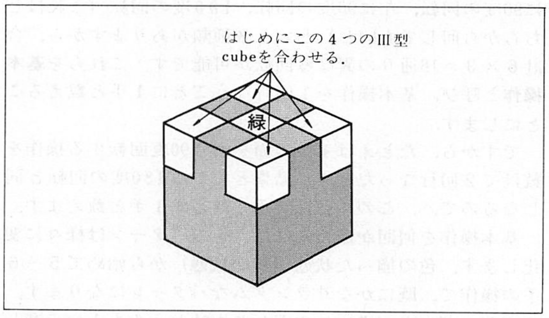
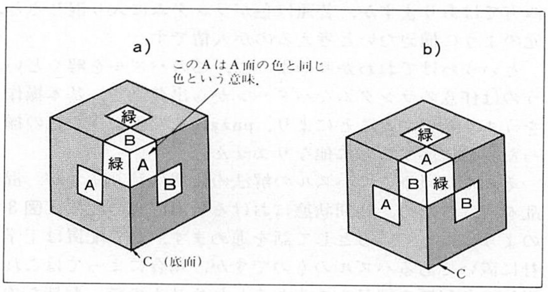


図8



A，C^{-1}，A^{-1}，C，A，C^{-1}，A^{-1}を行なえばよいのです(Cは時計回り，C^{-1}は反転時計回りです．正確には図9を参照してください)．以後このような対称的な手続きが出てきた場合は通常その一方を省略します．

緑の面のIII型の位置と向きが合いましたので，次は緑とは反対側にある黄色の面にくる残り4個のIII型の位置を合わせます．まず黄色の面を適当に回せば4個のうち1個の位置を合わすことができます．このままでは残り3個の位置が合っているとは限りませんが，隣り合う2個のIII型の位置を交換することができれば，それを組み合せてすべてのIII型の位置を合わすことができます．

隣り合う2つのIII型を交換する手続きを図10に示します．この手続きで緑のIII型cubeの情況は変化しません．この**基本手続き1**を高々3回(工夫すれば高々2回)適用することにより，III型の位置は完全に合います．

ところが，位置は合っても，まだ黄色の面のIII型cubeの向きが合っていませんから，最後にこれらの向きを合わせます．そのために図11の**基本手続き2**を使います．この基本手続きで，他のIII型cubeの情況を変えずに隣り合う2つのIII型cubeの向きを逆向きに回転することができますから，**基本手続き2**を高々3回適用することにより，最後の1つを除いてはIII型の向きが合います．最後の1つについては，他のIII型の向きを合わせれば自動的に向きが合うことが理論的に証明されています．

図9　図はpuzzleを斜め下から眺めたもの

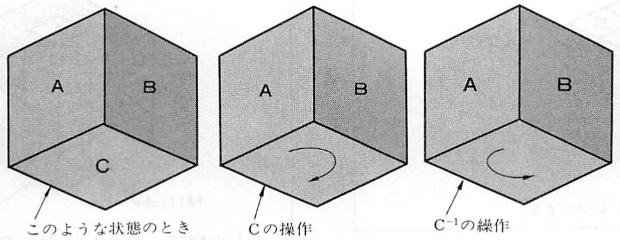
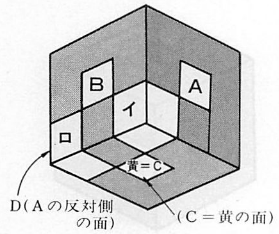
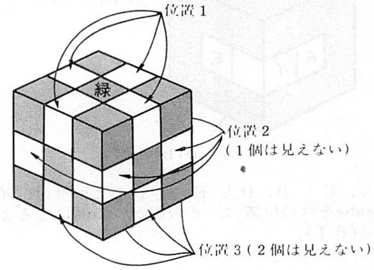


図10　基本手続き1



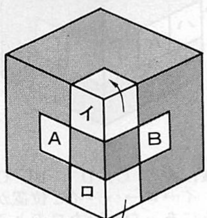
➡上の状態でB，C，B^{-1}，C^{-1}，D^{-1}，B^{-1}，Dを順に行なえばイ，ロのIII型cubeの位置を交換できる(7手)．

図12



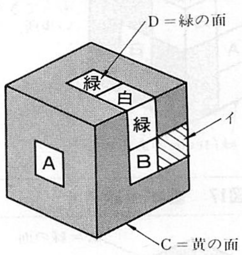
➡上の図で，位置1，2，3の順にII型cubeを合わせていく．

図11　基本手続き2



➡上の状態でまずA，B^{-1}という操作を3回繰り返し，次にA^{-1}，Bという操作を3回繰り返せば，III型の位置を変えずに，イのcubeを左に120度，ロのcubeを右に120度回転して向きを変えることができる(12手)．

図13



➡上の図でB，D，C^{-1}，A^{-1}，D^{-1}，C，の順に回わせば緑白のcubeをイの位置に移すことができる(6手)．

こうして，puzzleの各角にあるIII型cubeの位置と向きを完全に合わすことができました．

5　II型cubeを合わす

II型cubeを合わせる方法は，

❶緑の面にくる4個のII型の位置，向きを合わす．

❷緑の面のII型と隣り合う4個のII型の位置，向きを合わす．

❸緑の面とは反対側の黄色の面にくる4個のII型の位置，向きを合わす．

という3段階で進みます(図12)．

まず，緑の面にくるべき4個のcubeの合わせ方から考えます．緑の面にくるのは，どちらか一方の面が緑に塗られたcubeです．ここでは緑と白のcubeの合わせ方を紹介しますが，他の3個のII型cubeについてもまったく同様にして合わすことができます．

さて，緑白のcubeが図12の位置1，位置2，位置3のうちどこにあるかによって合わせ方は次の3つの場合に分れます．

《場合1》緑白cubeが位置1にある場合

初めから緑白の位置，向きが合っているならば何もする必要はありません．そうでないときは図13の手続きにより緑白cubeを位置2に移せるので，場合2に帰着します．

《場合2》緑白cubeが位置2にある場合

この場合は，緑の面を適当に回すことにより，緑白cubeを，図14のイまたはロの位置にもってくることができます．このとき，cubeの緑の面は面Aにきているようにします．

図14のイの位置にきた場合は，図15の基本手続き3を使って，緑白cubeを目的の位置に正しい向きで移せます．

図14のロの位置にきた場合は，図15の手続きと面対称な

図14

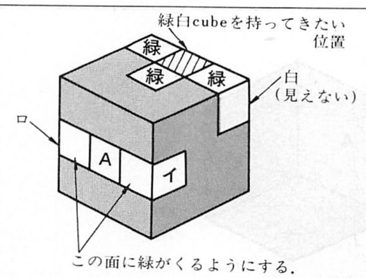
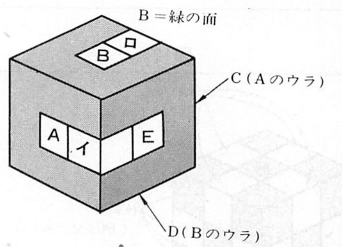


図15　基本手続き3



➡上の図で，C^{-1}，B，D^{-1}，E，B^{-1}，Dと回せば，イにあったcubeをロの位置に，イの面がロの面になるように移せる(6手).

図16

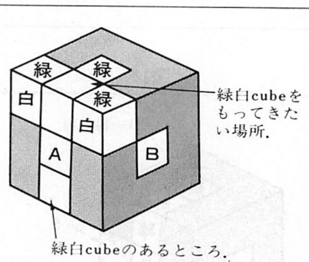
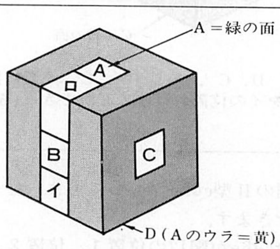
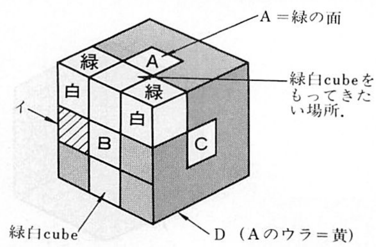


図17　基本手続き4



➡上の図で，B^{-1}，A^{-1}，D，C，A，D^{-1}の順に回すことにより，イのところのcubeをロの位置に，イの面がロの面にくるように移せる(6手).

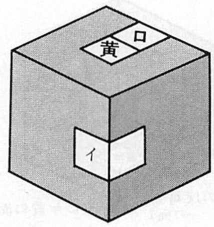
図18



上の図で，B，A^{-1}，D，C^{-1}，A，D^{-1}の順に回せば，緑白cubeをイの位置に持ってくることができる(6手).

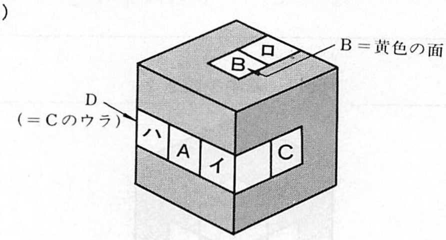
図19

a)



上げたい非黄cubeがイのところにくるようにおき，ロのところに黄cube(の1つ)がくるよう黄の面を回す.

b)



a)のようにした後，A^{-1}，C，D^{-1}，B^2，C^{-1}，D，A^{-1}と回す．すると，イ→ロ→ハ→イと位置が移り変わるので，イのところにあったcubeをロのところに上げることができる(7手)．必要があればそののちBを回してIII型cubeの位置を合わす.

手続きを行なえば移せるのでお試しください．なお，この段階でIII型の位置がずれることがありますが，緑の面を回せばすぐ直るので心配する必要はありません.

《場合3》緑白cubeが位置3にある場合

この場合は，緑の面を回すことにより，図16の状態にもっていくことができます．この時点で，緑白cubeの緑の面が図16のAの面を向いていれば，図17の**基本手続き4**により位置および向きを合わせることができます．そうでないときは，図18のようにすれば緑白cubeが位置2に移動しますから**場合2**に帰着します．**場合3**でも，III型の位置のずれが生じるので，そのときはやはり緑の面を回して修正します.

以上の3つの場合ですべての場合をつくしているので，どのような情況にあっても位置1のcubeは完全に合わすことができます．ところで，上の記述では緑と白を例に採用していますが，これらの言葉を聞いて思わず『ポン』と言ってしまう方は麻雀のやりすぎなのでご注意ください.

それでは次に，図12の位置2にくるべきcubeの位置と向きを合わせる段階へと進みます．緑の面は完全に合っているはずですから下にしておきます（当然黄色の面が上にきます)．ここで便宜上，残っている8個のII型cubeのうち，一方の面が黄色く塗られているcubeを**黄cube**，そうでないcubeを**非黄cube**と呼ぶことにします．これらは4個ずつあるのですが，位置2には非黄cubeを合わせるわけです.

これとは一見矛盾するように感じられるかも知れませんが，いったん非黄cubeをすべて位置3に上げてしまいます．これを行なう手続きを図19(a)，(b)に示します．実は，既に

図20

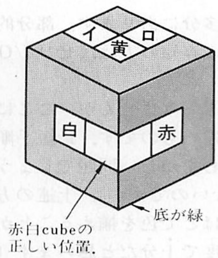
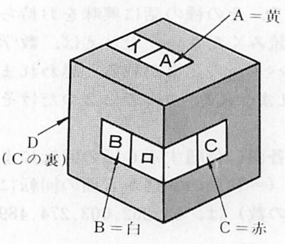
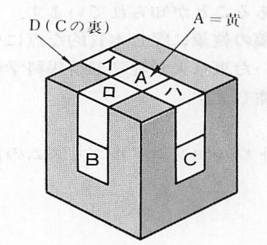


図21　基本手続き 5



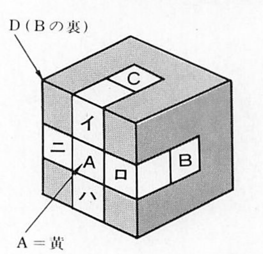
左の図で，D，A，B^2，A^2 B^2，A^2，B^2，A，D^{-1}と回せば，イの位置のcubeを，ロの位置に，イの面がロの面にくるように移すことができる（９手）．

図22　基本手続き 6



左の図で，B^2，A，D，C^{-1} B^2，D^{-1}，C，A，B^2と回せば，イ→ハ→ロ→イと巡回的にcubeを回せる（９手）．これと面対称な操作により，イ→ロ→ハ→イと回すこともできる．

図23　基本手続き 7



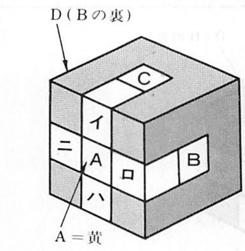
左の図で，B，D^{-1}，A，C^2 A^2，C^2，A^2，C^2，A，B^{-1}，Dと回せば，イ↔ハ，ニ↔ロ，とcubeをそれぞれ入れ替えることができる(11手)．

位置も向きも合っている非黄cubeは上げなくても済むのですが，簡単のため全部の非黄cubeを位置３に上げる方法をとりました．

この状態から位置２にくるべき非黄cubeの位置，向きを合わせるのですが，４個とも方法は同じですから，ここでは赤白cubeを例にとって説明します．初めに赤白cubeが図20のイまたはロの位置にくるように黄色の面を回します．このとき，赤白cubeの白の面が上だったならばイの位置に，赤の面が上だったならばロの位置にもってくるようにします．

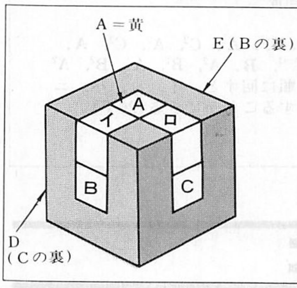
もし，白の面が上で赤白cubeをイの位置に回したならば**基本手続き５**（図21）を行なうと，赤白cubeを目的の位置に正しい向きで移すことができます．赤白cubeをロの位置に回した場合は**基本手続き５**と面対称な手続きを行なえば

図24　基本手続き 8



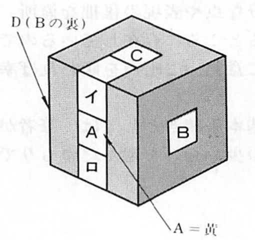
左の図で，B^{-1}，D^{-1}，C^{-1}，A^2，B^2，A^2，B^2，A^2，B^2，C，D，Bと回せば，イ↔ロ，ハ↔ニとcubeをそれぞれ入れ替えることができる(12手)．

図25　基本手続き 9



左の図で，B^2，A，C^{-1}，D，B^2，D^{-1}，C，A，B^2，C，D^{-1}，E^{-1}，D，C^{-1}，A^2，C D^{-1}，E^{-1}，D，C^{-1}の順に回すと，イ，ロのcubeの向きだけを反転することができる（20手）．

図26　基本手続き10



左の図で，B，D^{-1}，A，C^2 A^2，C^2，A^2，C^2，A，D，B^{-1}，C^{-1}，A，D^{-1}，A，C^2 A^2，C^2，A^2，C^2，A，D，A^{-1}，Cの順に回すと，イ，ロのcubeの向きだけを反転することができる(24手)．

うまくいくので各自お考えください．いずれにしてもIII型cubeの位置がずれることがあるので，その際は黄の面を回して元に戻します（手続きを誤っていなければ必ず元に戻るはずです）．

以上のような手続きを４回繰り返せば，位置２にくるべきcubeもすべて正しく揃えることができることになります．

いよいよ最後の位置３のcubeを合わせる段階に入ります．まず，残っている４個のII型cube（すべて黄cube）の位置がどのようにずれているかを確認します．実は，次の３つの可能性しかないことが理論的に証明されています．

❶３つのcubeが巡回的にずれている．
❷向かい合う２組のcubeが入れ替っている．
❸隣り合う２組のcubeが入れ替っている．

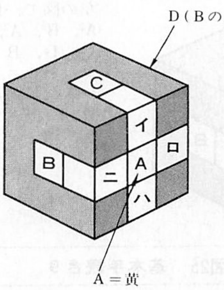
❶の場合の位置の合わせ方を図22に，❷の場合のものを図23に，❸の場合のものを図24にそれぞれ示します．これら３方法のいずれかを１回適用すれば，位置は正しくなりますが，向きが合っていないことがあります．向きの合わない可能性は，やはり理論的に次の３通りしか存在しません．

❶隣り合う２つのcubeの向きが反対．
❷向い合う２つのcubeの向きが反対．
❸４つとも向きが反対．



❶の場合の向きの合せ方を図25に，❷の場合を図26に，❸の場合を図27に示しました．これらの方法はいずれも手数が多いですから，誤りのないよう慎重に回してください．ここで間違うと一気に初めの状態に逆戻りしてしまいます．うまくいけば，パズルの色は見事に揃います．

図27　基本手続き 11



上の図で，D，B^{-1}，A，C^2，A^2，C^2，A^2，C^2，A，B，D^{-1}，C，B^{-1}，A，C^{-1}，B，A^2，B^2，A^2，B^2，A^2 B，C，A^{-1}，B，C^{-1}の順に回すと，イ，ロ，ハ，ニのcubeの向きだけを反転することができる(26手).

5 おわりに

以上，立方体パズルの解法についてだらだらと書き連ねてきましたが，記述の不充分な点や表現の稚拙な箇所，つまらぬ間違いが始んどいたるところに存在しているのではないかと心配しています．ご意見，ご叱声を賜われば幸いです．

また，解法を構成する各基本手続きとしては，筆者が現在知り得る限りの最も手順の少いものを選んだつもりですが，改良の余地は多分にあります．部分的あるいは全体的により良い方法をご存知の方は，ぜひI/O編集部までお寄せください．

なお，いうまでもありませんが，ここに示した方法は最短手順からはほど遠いものです．最短手順を求めることはこれよりはるかにむずかしい問題でしょう．しかしながら(言い逃れに過ぎないのですが)，上述の方法を使えば無器用な筆者でも10分ほどで色を揃えることができますから，実用的にはこの程度で十分だと思います（10分と十分とは洒落ではありません．偶然の一致です)．

それから，はじめのところでも述べたように，数学的な話は一切省いてしまったので，その種の話に興味をお持ちの方は，それなりの本をお読みください．たとえば，数学セミナー辺りには必ずこのパズルの記事が載ると思われます（既に出ているかも知れません)．ついでに2つだけそれらしい話をしましょう．

まず，立方体パズルで，各面に付随する6種の回転によって生成される群Gの位数（＝初めの状態から面の回転によって到達可能なパターンの数）は，43,252,003,274,489,856,000となります．

次に，Gを生成するのに本当に6種の回転が必要かどうかなのですが，これについては5種の回転（即ち1面の回転のみ禁止）が必要十分であることが知られています．

最後になりましたが，本稿の執筆に際し本質的な点について多くのご教示をいただいた東京大学理学部情報科学科の榎本彦衛先生に心から感謝します．

参考文献

1)エイホフロップ，クロフトウルマン：アルゴリズムの設計と解析，サイエンス

PC-8001 BASIC



ルービック・キューブ解法プログラム

■横澤信夫

ルービック・キューブを手にして回しているうちに，色がバラバラになり１色を揃えるのがやっとというとき，I/Oの'80年10月号の記事を読んで６色揃ったときのあの感激！

さっそく，手順を自分なりに紙にまとめて色のバラバラなキューブを回してみると，10分ほどで合わせられるようになりました．そこで，これをマイコンでやらせることはできないものかと考え，正月４日ほどつぶして作りあげたのがこのプログラムです．

実行の仕方

ルービック・キューブを手にして，考えながらいじっているうちに色がバラバラになり，元に戻らなくなったという人も多いと思います．

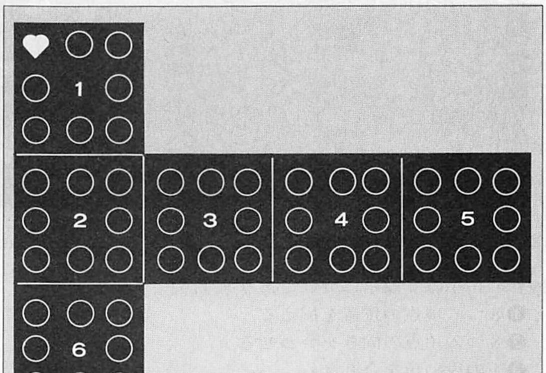
このプログラムは，マイコンにデータ（バラバラになった色の状態）を入力すれば，６面揃った状態になるように指示を出してくれるものです．色を合わせたいキューブをお持ちの方は用意してください．

プログラムをRUNさせると，説明が表示されます．色の合わせたいキューブをテーブルに置き，面と色にナンバーを付けます．テーブルに置いた状態で，上の面を１，手前の面を２，向かって右側の面を３，向こう側の面を４，向かって左側の面を５，下の面を６とし，図１のように展開して考えます．

次に各色にナンバーを付けるわけですが，色のナンバーは『各面の中央にある色をその面のナンバーと同じナンバーにする』と決めます．たとえば，上の面（No.１）の中央の色が黄のときは，黄はどこにあっても１というナンバーになるわけです．このようにして６色全部にナンバーを付けます．

面のナンバーと色のナンバーは，６色揃え終わるまで変えないでください．

図１　展開図



データの入力は図１の展開図が表示されるので，それを見ながらテーブルに置いてあるキューブの色の状態をテンキーで48個入力します．展開図に出ている『♥』の位置の色を間違えないよう１つ１つ入力してください．入力ミスがあるとやり直しを命じ，『ミスがある．nを押せ』と表示されます．

入力が終わると色を合わせるための指示が表示されます．

写真１　キューブの色の入力方法．

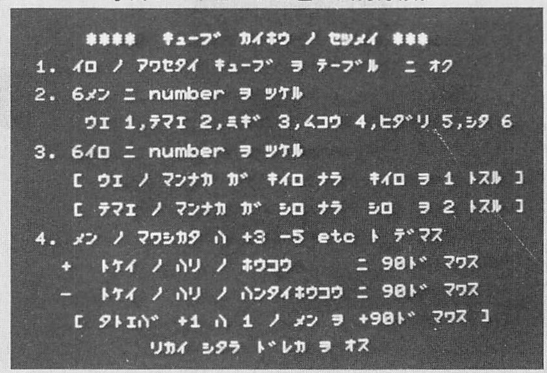


写真４　手順どおりまわして，角がそろいました．

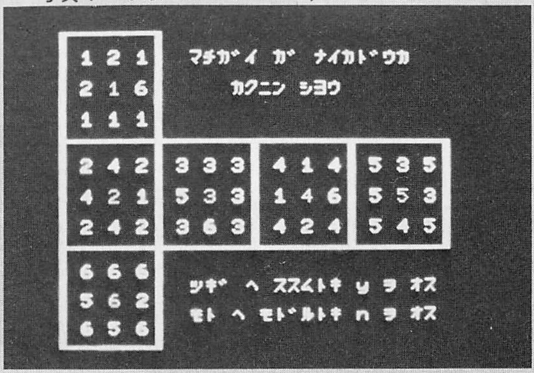


写真２　色を数字で，入力します．

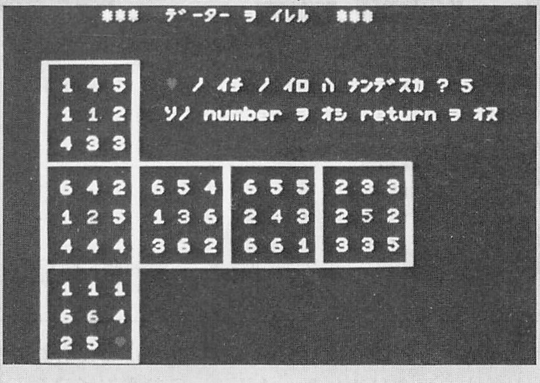


写真５　下の面がそろいました．

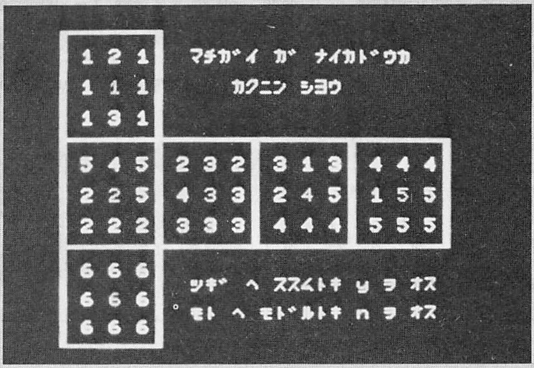


写真３　角の向きをあわせる手順

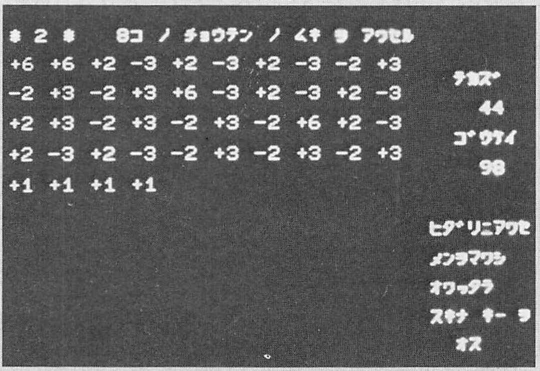
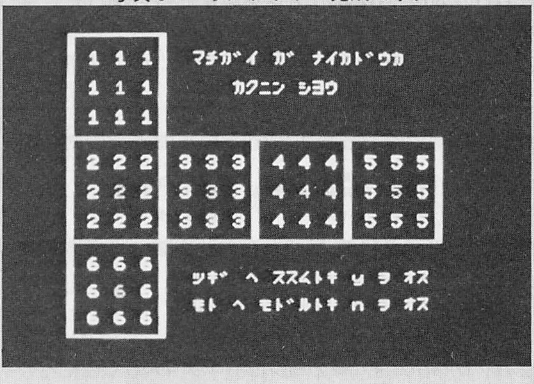


写真６　“オメデトウ”完成です．



回転の指示は５段階に分けてあります．

❶８つの頂点の位置を揃える．
❷８つの頂点の向きを合わせる．
❸下の段の色を合わせる．
❹中の段の色を合わせる．
❺上の段の色を合わせ，６面全部を揃える．

各段階とも終了した時点で，手数が何回かかるか表示されます．

表示は，

＋３，－４，－１，＋６，………

と出ますが，数字は各面のナンバーで，符号は90度回転させる方向の指示を表わします．『＋は時計の針の回る方向に90度回す』，『－は時計の針の回る方向と逆方向に90度回す』と考えるので，たとえば，『＋３』は，３の面を時計の針の方向に90度回すことになります．

このようにして，表示された回転の指示に従ってキューブの面を回転させていけば，５つの段階がすべて終わったとき，６面とも揃って買ったときの状態に戻ります．

ミスがあってうまくいかないときは，もう１度やり直してください．入力するときと回転させるときのミスが多いようですから，ゆっくり正確に実行してください．



プログラムの説明

このプログラムの手順はI/O'80年10月号の神谷慎吾氏の『ルービック・キューブの解き方』の手順を参考に作りました.

入力したデータの1の面はA1〜A8に，2の面はB1〜B8に，というように入れておき，回転させるたびにデータの移動と表示をさせるようにしました．各段階の手数は最高でも110を超えないようにしました.

また，各段階の終わりには，手数の表示のあと展開図も表示し，面の回転のさせ方にミスがなかったかどうか比較できるようにしてあります.

プログラムの大きな流れは表1のようになっています.

表1　プログラムの概要

行番号	内容
100〜120	初期値設定
130〜250	説明の表示
260〜480	データ入力部分
490〜630	面の回転と回転の表示のサブルーチン
640	各段階の手数の表示のサブルーチン
650〜830	展開図の表示のサブルーチン
840〜1100	8個の頂点の位置を合わせる
1110〜1290	8個の頂点の向きを合わせる
1300〜1590	下の段を合わせる
1600〜1950	中の段を合わせる
1960〜2320	上の段を合わせて色全部をそろえる
2330〜2340	完成したときの最後の表示

おわりに

このプログラムでは，ただ悲しいかな，どうしても人間の手数より多くなったり，無駄も時々やります．解法の最小手順をみつけるプログラムではないため，人間であればジッとみつめるとわかるものでも遠回りしてしまいます．でも，どのような状態からでもかならず6色そろえてくれます．試してみてください.

◘参考文献

1)神谷慎吾："ルービック・キューブの解き方", I/O, '80年10月号

2)数学セミナー，'80年11月号

3)N-BASICプログラム教本

ルービック・キューブ解法プログラム

```
100 REM ****  Rubik Cube ノ カイホウ  ****
110 REM ****        by ヨコサワ       ****
120 CONSOLE 0,25,0,1:WIDTH 40,25:COLOR 5,32,0:DIM N(7)
130 PRINTCHR$(12):COLOR5:PRINT "      ****  キューブ゛ カイホウ ノ セツメイ ***":PRINT
140 COLOR6:PRINT "1. イロ ノ アワセタイ キューブ゛ ヲ テーブ゛ル ニ オク":PRINT
150 PRINT "2. 6メン ニ number ヲ ツケル":PRINT
160 PRINT "     ウエ 1,テマエ 2,ミギ゛ 3,ムコウ 4,ヒダ゛リ 5,シタ 6":PRINT
170 PRINT "3. 6イロ ニ number ヲ ツケル":PRINT
180 PRINT "   [ ウエ ノ マンナカ カ゛ キイロ ナラ  キイロ ヲ 1 トスル ]"
190 PRINT "   [ テマエ ノ マンナカ カ゛ シロ ナラ  シロ  ヲ 2 トスル ]"
200 PRINT "4. メン ノ マワシカタ ハ +3 -5 etc ト デ゛マス":PRINT
210 PRINT "  +  トケイ ノ ハリ ノ ホウコウ     ニ 90ト゛ マワス":PRINT
220 PRINT" - トケイ ノ ハリ ノ ハンタイホウコウ ニ 90ト゛ マワス":PRINT
230 PRINT"  [ タトエハ゛ +1 ハ 1 ノ メン ヲ +90ト゛ マワス ]":PRINT
240 PRINT"         リカイ シタラ ト゛レカ ヲ オス"
250 X$=INKEY$:IF X$=""GOTO250
260 REM ***  データー ヲ イレル  ***
270 GOSUB670:GOSUB810:S=0:I=0:Z1=1:Z2=5
280 COLOR5:LOCATE0,0:PRINT "         ***  デ゛ーター ヲ イレル  ***"
290 COLOR2:LOCATE10,5:PRINT"●":COLOR7:LOCATE12,5:PRINT "ノ イチ ノ イロ ハ ナンデ゛スカ":LOCA
TE10,7:PRINT"ソノ number ヲ オシ return ヲ オス"
300 FOR P=1 TO 6
310 FOR K=1 TO 8
320 COLOR2:LOCATEZ1+1,Z2:PRINT"●";
330 COLOR7:LOCATE31,5:INPUT N(I):S=S+N(I)
340 COLOR7:LOCATEZ1,Z2:PRINT N(I);:I=I+1:GOSUB810
350 IF K<3 THEN Z1=Z1+2:GOTO390
360 IF K=3 THEN Z1=Z1-4:Z2=Z2+2:GOTO390
370 IF K=4 THEN Z1=Z1+4:GOTO390
380 IF K=5 THEN Z1=Z1-4:Z2=Z2+2:GOTO390 ELSE Z1=Z1+2
390 NEXT K
400 IF P=1 THEN Z1=1:Z2=12:A1=N(0):A2=N(1):A3=N(2):A4=N(3):A5=N(4):A6=N(5):A7=N(
6):A8=N(7)
410 IF P=2 THEN Z1=8:Z2=12:B1=N(0):B2=N(1):B3=N(2):B4=N(3):B5=N(4):B6=N(5):B7=N(
6):B8=N(7)
420 IF P=3 THEN Z1=15:Z2=12:C1=N(0):C2=N(1):C3=N(2):C4=N(3):C5=N(4):C6=N(5):C7=N
(6):C8=N(7)
430 IF P=4 THEN Z1=22:Z2=12:D1=N(0):D2=N(1):D3=N(2):D4=N(3):D5=N(4):D6=N(5):D7=N
(6):D8=N(7)
440 IF P=5 THEN Z1=1:Z2=19:E1=N(0):E2=N(1):E3=N(2):E4=N(3):E5=N(4):E6=N(5):E7=N(
6):E8=N(7)
450 I=0:NEXTP:F1=N(0):F2=N(1):F3=N(2):F4=N(3):F5=N(4):F6=N(5):F7=N(6):F8=N(7)
460 GOSUB650:IF S=168 GOTO470 ELSE LOCATE30,0:COLOR2:PRINT"ミス カ゛ アル":LOCATE30,2:
PRINT"n ヲ オセ":COLOR6
470 X$=INKEY$:IF X$="n"ORX$="y"GOTO480 ELSE GOTO470
480 IF X$="n"GOTO260 ELSE 840
490 REM *** カイテン ***
500 A0=A1:A1=A6:A6=A8:A8=A3:A3=A0:A0=A2:A2=A4:A4=A7:A7=A5:A5=A0:A0=B1:B1=C1:C1=D
1:D1=E1:E1=A0:A0=B2:B2=C2:C2=D2:D2=E2:E2=A0:A0=B3:B3=C3:C3=D3:D3=E3:E3=A0:T$="+1
```

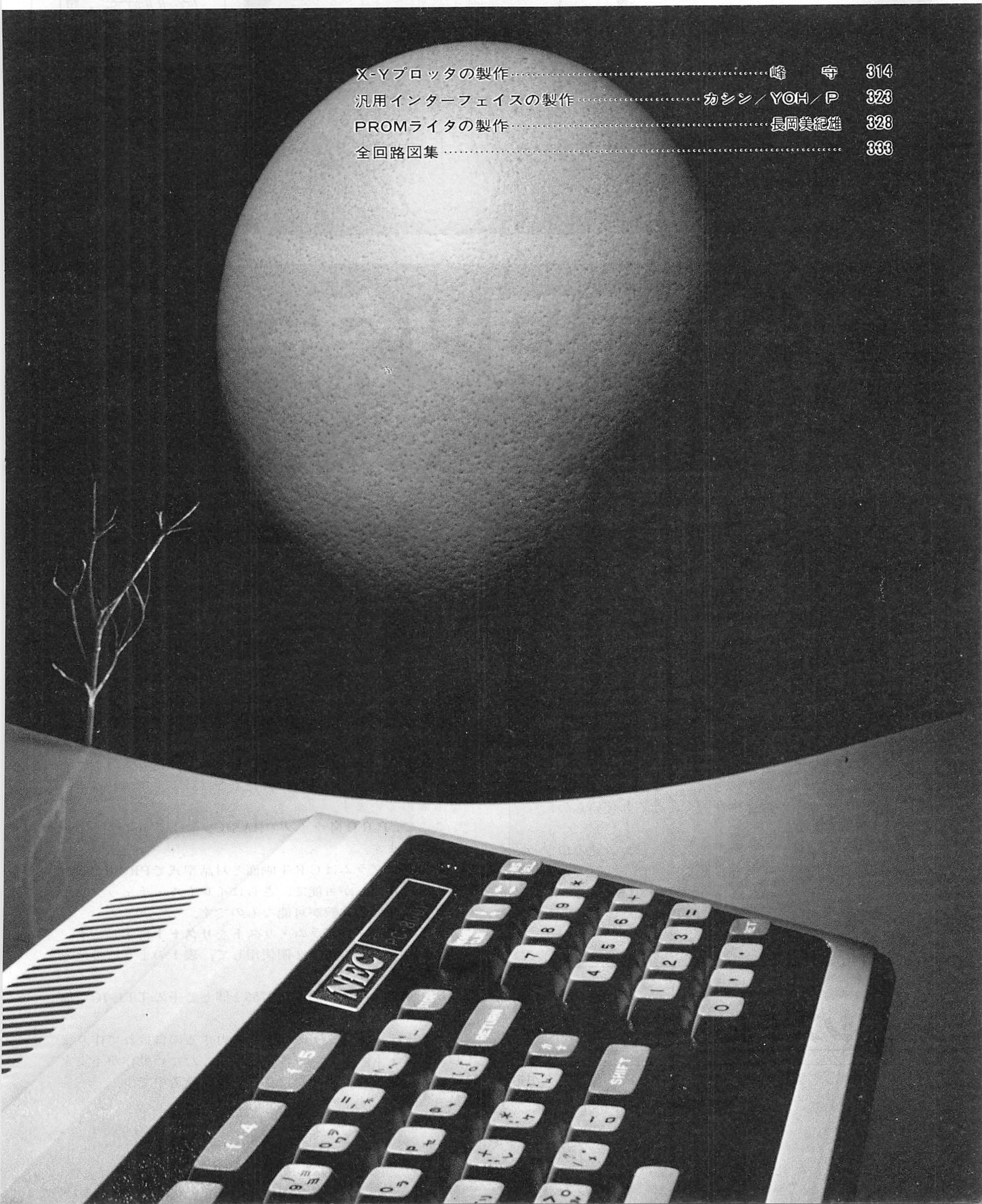
```
":P1=P1+1:GOTO620
510 A0=A1:A1=A3:A3=A8:A8=A6:A6=A0:A0=A2:A2=A5:A5=A7:A7=A4:A4=A0:A0=B1:B1=E1:E1=D
1:D1=C1:C1=A0:A0=B2:B2=E2:E2=D2:D2=C2:C2=A0:A0=B3:B3=E3:E3=D3:D3=C3:C3=A0:T$="-1
":P1=P1+1:GOTO620
520 B0=B1:B1=B6:B6=B8:B8=B3:B3=B0:B0=B2:B2=B4:B4=B7:B7=B5:B5=B0:B0=C1:C1=A6:A6=E
8:E8=F3:F3=B0:B0=C6:C6=A8:A8=E3:E3=F1:F1=B0:B0=C4:C4=A7:A7=E5:E5=F2:F2=B0:T$="+2
":P1=P1+1:GOTO620
530 B0=B1:B1=B3:B3=B8:B8=B6:B6=B0:B0=B2:B2=B5:B5=B7:B7=B4:B4=B0:B0=C1:C1=F3:F3=E
8:E8=A6:A6=B0:B0=C6:C6=F1:F1=E3:E3=A8:A8=B0:B0=C4:C4=F2:F2=E5:E5=A7:A7=B0:T$="-2
":P1=P1+1:GOTO620
540 C0=C1:C1=C6:C6=C8:C8=C3:C3=C0:C0=C2:C2=C4:C4=C7:C7=C5:C5=C0:C0=D1:D1=A8:A8=B
8:B8=F8:F8=C0:C0=D6:D6=A3:A3=B3:B3=F3:F3=C0:C0=D4:D4=A5:A5=B5:B5=F5:F5=C0:T$="+3
":P1=P1+1:GOTO620
550 C0=C1:C1=C3:C3=C8:C8=C6:C6=C0:C0=C2:C2=C5:C5=C7:C7=C4:C4=C0:C0=D1:D1=F8:F8=B
8:B8=A8:A8=C0:C0=F3:F3=B3:B3=A3:A3=D6:D6=C0:C0=F5:F5=B5:B5=A5:A5=D4:D4=C0:T$="-3
":P1=P1+1:GOTO620
560 D0=D1:D1=D6:D6=D8:D8=D3:D3=D0:D0=D2:D2=D4:D4=D7:D7=D5:D5=D0:D0=E1:E1=A3:A3=C
8:C8=F6:F6=D0:D0=E6:E6=A1:A1=C3:C3=F8:F8=D0:D0=E4:E4=A2:A2=C5:C5=F7:F7=D0:T$="+4
":P1=P1+1:GOTO620
570 D0=D1:D1=D3:D3=D8:D8=D6:D6=D0:D0=D2:D2=D5:D5=D7:D7=D4:D4=D0:D0=E1:E1=F6:F6=C
8:C8=A3:A3=D0:D0=E6:E6=F8:F8=C3:C3=A1:A1=D0:D0=E4:E4=F7:F7=C5:C5=A2:A2=D0:T$="-4
":P1=P1+1:GOTO620
580 E0=E1:E1=E6:E6=E8:E8=E3:E3=E0:E0=E2:E2=E4:E4=E7:E7=E5:E5=E0:E0=B1:B1=A1:A1=D
8:D8=F1:F1=E0:E0=B6:B6=A6:A6=D3:D3=F6:F6=E0:E0=B4:B4=A4:A4=D5:D5=F4:F4=E0:T$="+5
":P1=P1+1:GOTO620
590 E0=E1:E1=E3:E3=E8:E8=E6:E6=E0:E0=E2:E2=E5:E5=E7:E7=E4:E4=E0:E0=B1:B1=F1:F1=D
8:D8=A1:A1=E0:E0=B4:B4=F4:F4=D5:D5=A4:A4=E0:E0=B6:B6=F6:F6=D3:D3=A6:A6=E0:T$="-5
":P1=P1+1:GOTO620
600 F0=F1:F1=F6:F6=F8:F8=F3:F3=F0:F0=F2:F2=F4:F4=F7:F7=F5:F5=F0:F0=B6:B6=E6:E6=D
6:D6=C6:C6=F0:F0=B8:B8=E8:E8=D8:D8=C8:C8=F0:F0=B7:B7=E7:E7=D7:D7=C7:C7=F0:T$="+6
":P1=P1+1:GOTO620
610 F0=F1:F1=F3:F3=F8:F8=F6:F6=F0:F0=F2:F2=F5:F5=F7:F7=F4:F4=F0:F0=B6:B6=C6:C6=D
6:D6=E6:E6=F0:F0=B8:B8=C8:C8=D8:D8=E8:E8=F0:F0=B7:B7=C7:C7=D7:D7=E7:E7=F0:T$="-6
":P1=P1+1:GOTO620
620 T1=T1+1:IF T1=11 THEN T1=T1-10:T2=T2+2
630 LOCATE T1*3-3,T2+2:COLOR6:PRINT T$:RETURN
640 COLOR7:LOCATE33,3:PRINT "ﾃｶｽﾞ":LOCATE34,5:PRINT P1:LOCATE33,7:PRINT "ｺﾞｳｹｲ":
LOCATE34,9:PRINT P2:P1=0:LOCATE31,13:PRINT"ﾋﾀﾞﾘﾆｱﾜｾ":LOCATE31,15:PRINT"ﾒﾝｦﾏﾜｼ":L
OCATE31,17:PRINT "ｵﾜｯﾀﾗ":LOCATE31,19:PRINT"ｽｷﾅ ｷｰ ｦ":LOCATE33,21:PRINT"ｵｽ":RETUR
N
650 REM *** ﾋｮｳｼﾞ ***
660 GOSUB670:GOSUB740:GOSUB810:RETURN
670 PRINTCHR$(12)
680 COLOR3:LOCATE 4,7:PRINT "1"
690 LOCATE 4,14:PRINT "2"
700 LOCATE 11,14:PRINT "3"
710 LOCATE 18,14:PRINT "4"
720 LOCATE 25,14:PRINT "5"
730 LOCATE 4,21:PRINT "6":RETURN
740 COLOR7:LOCATE1,5:PRINTA1:LOCATE3,5:PRINTA2:LOCATE5,5:PRINTA3:LOCATE1,7:PRINT
A4:LOCATE5,7:PRINTA5:LOCATE1,9:PRINTA6:LOCATE3,9:PRINTA7:LOCATE5,9:PRINTA8
750 LOCATE1,12:PRINTB1:LOCATE3,12:PRINTB2:LOCATE5,12:PRINTB3:LOCATE1,14:PRINTB4:
LOCATE5,14:PRINTB5:LOCATE1,16:PRINTB6:LOCATE3,16:PRINTB7:LOCATE5,16:PRINTB8
760 LOCATE8,12:PRINTC1:LOCATE10,12:PRINTC2:LOCATE12,12:PRINTC3:LOCATE8,14:PRINTC
4:LOCATE12,14:PRINTC5:LOCATE8,16:PRINTC6:LOCATE10,16:PRINTC7:LOCATE12,16:PRINTC8

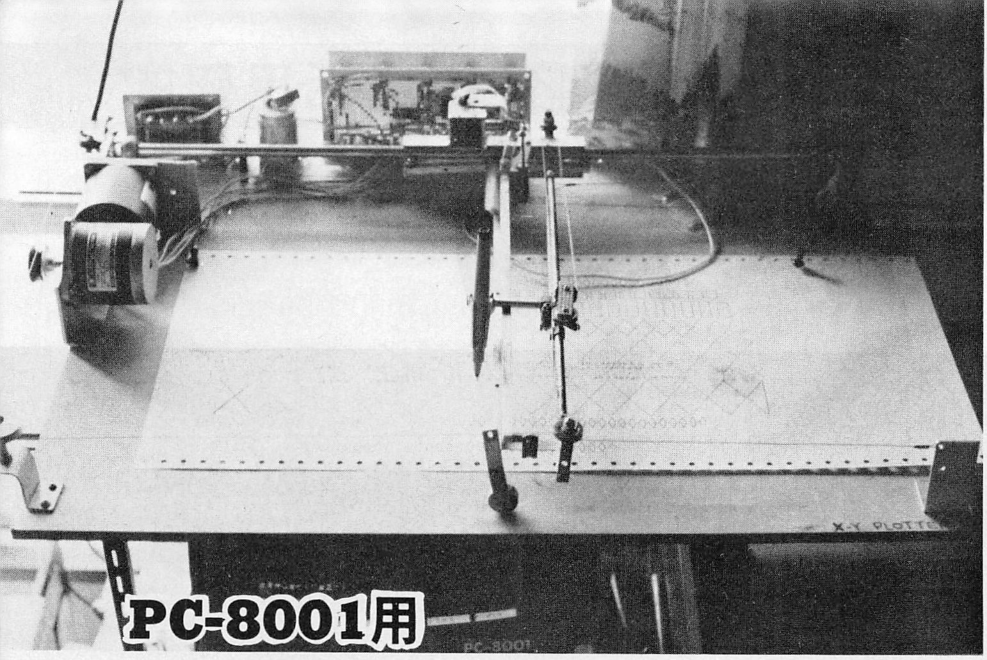
770 LOCATE15,12:PRINTD1:LOCATE17,12:PRINTD2:LOCATE19,12:PRINTD3:LOCATE15,14:PRIN
TD4:LOCATE19,14:PRINTD5:LOCATE15,16:PRINTD6:LOCATE17,16:PRINTD7:LOCATE19,16:PRIN
TD8
780 LOCATE22,12:PRINTE1:LOCATE24,12:PRINTE2:LOCATE26,12:PRINTE3:LOCATE22,14:PRIN
TE4:LOCATE26,14:PRINTE5:LOCATE22,16:PRINTE6:LOCATE24,16:PRINTE7:LOCATE26,16:PRIN
TE8
790 LOCATE1,19:PRINTF1:LOCATE3,19:PRINTF2:LOCATE5,19:PRINTF3:LOCATE1,21:PRINTF4:
LOCATE5,21:PRINTF5:LOCATE1,23:PRINTF6:LOCATE3,23:PRINTF7:LOCATE5,23:PRINTF8
800 COLOR5:LOCATE10,5:PRINT"ﾏﾁｶﾞｲ ｶﾞ ﾅｲｶﾄﾞｳｶ":LOCATE13,7:PRINT"ｶｸﾆﾝ ｼﾖｳ":LOCATE1
0,20:PRINT "ﾂｷﾞ ﾍ ｽｽﾑﾄｷ ｯ ｦ ｵｽ":LOCATE10,22:PRINT "ﾓﾄ ﾍ ﾓﾄﾞﾙﾄｷ n ｦ ｵｽ":RETURN
810 COLOR6:LINE(1,15)-(15,99),PSET,B
820 LINE(1,43)-(57,71),PSET,B
830 LINE(29,43)-(43,71),PSET,B:RETURN
840 REM ***  8ｺ ﾉ ﾁｮｵﾃﾝ ﾉ ｲﾁ ｦ ｱﾜｾﾙ  ***
850 PRINTCHR$(12):T1=0:T2=0:P1=0:P2=0
860 COLOR5:LOCATE0,0:PRINT "* 1 *   8ｺ ﾉ ﾁｮｵﾃﾝ ﾉ ｲﾁ ｦ ｱﾜｾﾙ"
870 IF(A6=1ORE3=1ORB1=1)AND(A8=1ORB3=1ORC1=1)AND(A3=1ORC3=1ORD1=1)AND(A1=1ORD3=1
ORE1=1)GOTO980
880 IF A8=1ORB3=1ORC1=1 THEN GOSUB500:GOSUB500:GOSUR590:GOTO900 ELSE GOSUB530
890 IF A8=1ORB3=1ORC1=1 THEN GOSUB500:GOSUB500:GOSUB590:GOTO900 ELSE GOSUB540:GO
TO880
900 IF A8=1ORB3=1ORC1=1 THEN GOSUB500:GOSUB500:GOTO920 ELSE GOSUB530
910 IF A8=1ORB3=1ORC1=1 THEN GOSUB500:GOSUB500:GOTO920 ELSE GOSUB540:GOTO900
920 IF A8=1ORB3=1ORC1=1 THEN GOSUB530:GOTO940 ELSE GOSUB530
930 IF A8=1ORB3=1ORC1=1 THEN GOSUB530:GOTO940 ELSE GOSUB540:GOTO930
940 IF A8=1ORB3=1ORC1=1 GOTO970 ELSE GOSUB530
950 IF A8=1ORB3=1ORC1=1 GOTO960 ELSE GOSUB540:GOTO950
960 GOSUB540:GOSUB520:GOSUB550
970 GOSUB570
980 IF (B6=2ORE8=2ORF1=2)AND(B6=5ORE8=5ORF1=5) GOTO990 ELSE GOSUB600:GOTO980
```

```
990 K1=B8+C6+F3:IF K1=11 GOTO1010 ELSE GOSUB540:GOSUB600:GOSUB550:GOSUB610:GOSUB
530:GOSUB550:GOSUB520
1000 K1=B8+C6+F3:IF K1=11 GOTO1010 ELSE GOSUB560:GOSUB600:GOSUB570:GOSUB610:GOSU
B550:GOSUB570:GOSUB540:GOTO990
1010 K1=C8+D6+F8:IF K1=13 GOTO1020 ELSE GOSUB560:GOSUB600:GOSUB570:GOSUB610:GOSU
B550:GOSUB570:GOSUB540:GOTO1010
1020 IF(A6=2ORB1=2ORE3=2)AND(A6=5ORB1=5ORE3=5) GOTO1030 ELSE GOSUB500:GOTO1020
1030 K1=A8+B3+C1:IF K1=6 GOTO1050 ELSE GOSUB540:GOSUB500:GOSUB550:GOSUB510:GOSUB
570:GOSUB550:GOSUB560
1040 K1=A8+B3+C1:IF K1=6 GOTO1050 ELSE GOSUB560:GOSUB500:GOSUB570:GOSUB510:GOSUB
590:GOSUB570:GOSUB580:GOTO1030
1050 K1=A3+C3+D1:IF K1=8 GOTO1060 ELSE GOSUB560:GOSUB500:GOSUB570:GOSUB510:GOSUB
590:GOSUB570:GOSUB580:GOTO1050
1060 P2=P2+P1:GOSUB640
1070 X$=INKEY$:IF X$=""GOTO1070
1080 GOSUB650
1090 X$=INKEY$:IF X$="n"ORX$="y"GOTO1100 ELSE GOTO1090
1100 IF X$="n"GOTO260
1110 REM *** 8ｺ ﾉ ﾁｮｳﾃﾝ ﾉ ﾑｷ ｦ ｱﾜｾﾙ ***
1120 PRINTCHR$(12):T1=0:T2=0
1130 COLOR5:LOCATE0,0:PRINT "* 2 *   8ｺ ﾉ ﾁｮｳﾃﾝ ﾉ ﾑｷ ｦ ｱﾜｾﾙ"
1140 IF A1=1ANDA3=1ANDA6=1ANDA8=1ANDF1=6ANDF3=6ANDF6=6ANDF8=6 GOTO1250
1150 FOR K=1 TO 4
1160 GOSUB600:IF F3=6 GOTO1190
1170 IF B8=6 THEN GOSUB550:GOSUB520:GOSUB550:GOSUB520:GOSUB550:GOSUB520:GOSUB540
:GOSUB530:GOSUB540:GOSUB530:GOSUB540:GOSUB530:GOTO1190
1180 GOSUB520:GOSUB550:GOSUB520:GOSUB550:GOSUB520:GOSUB550:GOSUB530:GOSUB540:GOS
UB530:GOSUB540:GOSUB530:GOSUB540
1190 NEXT
1200 FOR K=1 TO 4
1210 GOSUB500:IF A6=1 GOTO1240
1220 IF E3=1 THEN GOSUB500:GOSUB530:GOSUB500:GOSUB530:GOSUB500:GOSUB530:GOSUB510
:GOSUB520:GOSUB510:GOSUB520:GOSUB510:GOSUB520:GOTO1240
1230 GOSUB530:GOSUB500:GOSUB530:GOSUB500:GOSUB530:GOSUB500:GOSUB520:GOSUB510:GOS
UB520:GOSUB510:GOSUB520:GOSUB510
1240 NEXT
1250 P2=P2+P1:GOSUB640
1260 X$=INKEY$:IF X$=""GOTO1260
1270 GOSUB650
1280 X$=INKEY$:IF X$="n"ORX$="y"GOTO1290 ELSE GOTO1280
1290 IF X$="n"GOTO260
1300 REM *** ｼﾀ ﾉ ｲﾛ ｦ ｿﾛｴﾙ ***
1310 PRINTCHR$(12):T1=0:T2=0
1320 COLOR5:LOCATE0,0:PRINT "* 3 *   ｼﾀ ﾉ ｲﾛ ｦ ｿﾛｴﾙ"
1330 IF F2=6ANDF4=6ANDF5=6ANDF7=6ANDB7=2ANDC7=3ANDD7=4ANDE7=5 GOTO1530
1340 FOR K=1 TO .4
1350 IF B7<>6ANDF2<>6 THEN GOSUB600:GOTO1370
1360 IF A7=6ORB2=6 THEN GOSUB500:GOTO1360 ELSE GOSUB520:GOSUB510:GOSUB600:GOSUB5
50
1370 NEXT
1380 FOR K=1 TO 2
1390 IF B4=6ORE5=6 GOTO1400 ELSE GOTO1410
1400 IF A5=6ORC2=6 THEN GOSUB500:GOTO1400 ELSE GOSUB590:GOSUB520:GOSUB570:GOSUB5
00:GOSUB500:GOSUB530:GOSUB560:GOSUB590
1410 IF E4=6ORD5=6 GOTO1420 ELSE GOTO1430
1420 IF A7=6ORB2=6 THEN GOSUB500:GOTO1420 ELSE GOSUB570:GOSUB580:GOSUB550:GOSUB5
00:GOSUB500:GOSUB590:GOSUB540:GOSUB570
1430 IF D4=6ORC5=6 GOTO1440 ELSE GOTO1450
1440 IF A4=6ORE2=6 THEN GOSUB500:GOTO1440 ELSE GOSUB550:GOSUB560:GOSUB530:GOSUB5
00:GOSUB500:GOSUB570:GOSUB520:GOSUB550
1450 IF C4=6ORB5=6 GOTO1460 ELSE GOTO1470
1460 IF A2=6ORD2=6 THEN GOSUB500:GOTO1460 ELSE GOSUB530:GOSUB540:GOSUB590:GOSUB5
00:GOSUB500:GOSUB550:GOSUB580:GOSUB530
1470 NEXT
1480 FOR K=1 TO 4
1490 IF(A7=B6ORB2=B6)AND(A7=6ORB2=6)GOTO1500 ELSE GOSUB500:GOTO1490
1500 IF A7=B6 THEN GOSUB1580 ELSE GOSUB1590
1510 NEXT
1520 IF B1<>5 THEN GOSUB500:GOTO1520
1530 P2=P2+P1:GOSUB640
1540 X$=INKEY$:IF X$=""GOTO1540
1550 GOSUB650
1560 X$=INKEY$:IF X$="n"ORX$="y"GOTO1570 ELSE GOTO1560
1570 IF X$="n"GOTO260 ELSE GOTO1600
1580 GOSUB520:GOSUB510:GOSUB600:GOSUB550:RETURN
1590 GOSUB530:GOSUB510:GOSUB600:GOSUB540:GOSUB540:GOSUB500:GOSUB610:GOSUB530:GOS
UB600:RETURN
1600 REM *** ﾅｶﾉﾀﾞﾝ ﾉ ｲﾛ ｦ ｱﾜｾﾙ ***
1610 PRINTCHR$(12):T1=0:T2=0
1620 COLOR5:LOCATE0,0:PRINT "* 4 * ﾅｶﾉﾀﾞﾝ ﾉ ｲﾛ ｦ ｱﾜｾﾙ"
1630 IF B4=2ANDB5=2ANDC4=3ANDC5=3ANDD4=4ANDD5=4ANDE4=5ANDE5=5 GOTO1830
1640 FOR K=1 TO 2
1650 IF B4<>1ANDE5<>1 GOTO1660 ELSE GOTO1670
1660 IF A5<>1ANDC2<>1 THEN GOSUB500:GOTO1660 ELSE GOSUB590:GOSUB520:GOSUB570:GOS
UB500:GOSUB500:GOSUB530:GOSUB560:GOSUB590
1670 IF E4<>1ANDD5<>1 GOTO1680 ELSE GOTO1690
```

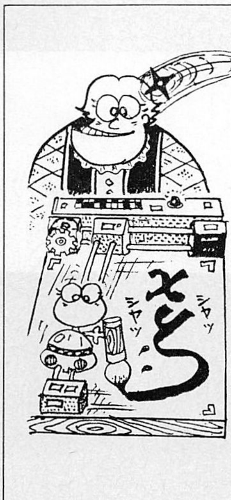
```
1680 IF A7<>1ANDB2<>1 THEN GOSUB500:GOTO1680 ELSE GOSUB570:GOSUB580:GOSUB550:GOS
UB500:GOSUB500:GOSUB590:GOSUB540:GOSUB570
1690 IF D4<>1ANDC5<>1 GOTO1700 ELSE GOTO1710
1700 IF A4<>1ANDE2<>1 THEN GOSUB500:GOTO1700 ELSE GOSUB550:GOSUB560:GOSUB530:GOS
UB500:GOSUB500:GOSUB570:GOSUB520:GOSUB550
1710 IF C4<>1ANDB5<>1 GOTO1720 ELSE GOTO1730
1720 IF A2<>1ANDD2<>1 THEN GOSUB500:GOTO1720 ELSE GOSUB530:GOSUB540:GOSUB590:GOS
UB500:GOSUB500:GOSUB550:GOSUB580:GOSUB530
1730 NEXT
1740 IF (A4=2ORE2=2)AND(A4=3ORE2=3) GOTO1750 ELSE GOSUB500:GOTO1740
1750 IF A4=2 THEN GOSUB580:GOSUB1880:GOSUB590 ELSE GOSUB500:GOSUB570:GOSUB1890:G
OSUB560
1760 IF (A7=3ORB2=3)AND(A7=4ORB2=4) GOTO1770 ELSE GOSUB500:GOTO1760
1770 IF A7=3 THEN GOSUB520:GOSUB1900:GOSUB530 ELSE GOSUB500:GOSUB590:GOSUB1910:G
OSUB580
1780 IF (A5=4ORC2=4)AND(A5=5ORC2=5) GOTO1790 ELSE GOSUB500:GOTO1780
1790 IF A5=4 THEN GOSUB540:GOSUB1920:GOSUB550 ELSE GOSUB500:GOSUB530:GOSUB1930:G
OSUB520
1800 IF (A2=5ORD2=5)AND(A2=2ORD2=2) GOTO1810 ELSE GOSUB500:GOTO1800
1810 IF A2=5 THEN GOSUB560:GOSUB1940:GOSUB570 ELSE GOSUB500:GOSUB550:GOSUB1950:G
OSUB540
1820 IF B1=2 GOTO1830 ELSE GOSUB500:GOTO1820
1830 P2=P2+P1:GOSUB640
1840 X$=INKEY$:IF X$=""GOTO1840
1850 GOSUB650
1860 X$=INKEY$:IF X$="n"ORX$="y"GOTO1870 ELSE GOTO1860
1870 IF X$="n"GOTO260 ELSE GOTO1960
1880 FOR K=1 TO 3:GOSUB500:GOSUB520:GOSUB520:GOSUB500:NEXT:RETURN
1890 FOR K=1 TO 3:GOSUB510:GOSUB550:GOSUB550:GOSUB510:NEXT:RETURN
1900 FOR K=1 TO 3:GOSUB500:GOSUB540:GOSUB540:GOSUB500:NEXT:RETURN
1910 FOR K=1 TO 3:GOSUB510:GOSUB570:GOSUB570:GOSUB510:NEXT:RETURN
1920 FOR K=1 TO 3:GOSUB500:GOSUB560:GOSUB560:GOSUB500:NEXT:RETURN
1930 FOR K=1 TO 3:GOSUB510:GOSUB590:GOSUB590:GOSUB510:NEXT:RETURN
1940 FOR K=1 TO 3:GOSUB500:GOSUB580:GOSUB580:GOSUB500:NEXT:RETURN
1950 FOR K=1 TO 3:GOSUB510:GOSUB530:GOSUB530:GOSUB510:NEXT:RETURN
1960 REM *** ｾﾞﾝﾌﾞﾉｲﾛ ｦ ｿﾛｴﾙ ***
1970 PRINTCHR$(12):T1=0:T2=0:L1=0:L2=0
1980 COLOR5:LOCATE0,0:PRINT "* 5 *   ｾﾞﾝﾌﾞﾉｲﾛ ｦ ｿﾛｴﾙ"
1990 IF B1=B2+A7-1 GOTO2000 ELSE L1=L1+1:L2=L2+B1
2000 IF C1=C2+A5-1 GOTO2010 ELSE L1=L1+1:L2=L2+C1
2010 IF D1=D2+A2-1 GOTO2020 ELSE L1=L1+1:L2=L2+D1
2020 IF E1=E2+A4-1 GOTO2030 ELSE L1=L1+1:L2=L2+E1
2030 IF L1=0 GOTO2110
2040 IF L1=3 GOTO2050 ELSE GOTO2080
2050 IF L2=9 GOTO2070
2060 FOR K1=1 TO L2-9:GOSUB510:NEXT
2070 GOSUB540:GOSUB540:GOSUB500:GOSUB520:GOSUB570:GOSUB540:GOSUB540:GOSUB530:GOS
UB560:GOSUB500:GOSUB540:GOSUB540:IF B1=B2+A7-1 GOTO2110 ELSE GOTO2070
2080 IF D1=B2+A7-1 THEN GOSUB520:GOSUB570:GOSUB1900:GOSUB530:GOSUB560:GOTO2110
2090 IF B1=A4+E2-1 THEN GOSUB500
2100 GOSUB530:GOSUB570:GOSUB550:GOSUB500:GOSUB1880:GOSUB510:GOSUB540:GOSUB560:GO
SUB520
2110 IF B1<>2 THEN GOSUB500:GOTO2110
2120 L1=0:L2=0
2130 IF B1=B2 GOTO2140 ELSE L1=L1+1:L2=L2+B1
2140 IF C1=C2 GOTO2150 ELSE L1=L1+1:L2=L2+C1
2150 IF D1=D2 GOTO2160 ELSE L1=L1+1:L2=L2+D1
2160 IF E1=E2 GOTO2170 ELSE L1=L1+1:L2=L2+E1+1
2170 IF L1=0 GOTO2270
2180 IF L1=2 GOTO2190 ELSE GOSUB2260:GOSUB500:GOSUB2260:GOTO2270
2190 IF L2=6ORL2=9 GOTO2200 ELSE GOTO2220
2200 IF L2=6 THEN GOSUB500
2210 GOSUB2260:GOTO2270
2220 IF L2=8 THENL2=L2+4
2230 IF L2=5 GOTO2250 ELSE L3=FIX((L2-5)/2)
2240 FOR K=1 TO L3:GOSUB500:NEXT
2250 GOSUB520:GOSUB520:GOSUB500:GOSUB550:GOSUB580:GOSUB520:GOSUB520:GOSUB590:GOS
UB540:GOSUB500:GOSUB520:GOSUB520:GOSUB540:GOSUB590:GOSUB570:GOSUB580:GOSUB550:GO
SUB500:GOSUB500:GOSUB540:GOSUB590:GOSUB570:GOSUB580:GOSUB550:GOTO2270
2260 GOSUB520:GOSUB570:GOSUB1900:GOSUB560:GOSUB530:GOSUB550:GOSUB500:GOSUB570:GO
SUB1900:GOSUB560:GOSUB510:GOSUB540:RETURN
2270 IF B1<>2 THEN GOSUB500:GOTO2270
2280 P2=P2+P1:GOSUB640
2290 X$=INKEY$:IF X$=""GOTO2290
2300 GOSUB650
2310 X$=INKEY$:IF X$="n"ORX$="y"GOTO2320 ELSE GOTO2310
2320 IF X$="n"GOTO260
2330 PRINTCHR$(12):LOCATE5,5:COLOR6:PRINT "***  ｲﾛ ｶﾞ ｱｯﾃ ｵﾒﾃﾞﾄｳ  ***":LOCATE10,
15:COLOR7:PRINT "[ y ｦ ｵｽﾄ ﾏﾀ ﾊｼﾞﾒ ﾆ ﾓﾄﾞﾙ ]"
2340 X$=INKEY$:IF X$="y"GOTO130 ELSE GOTO2340
```

ハード編





PC-8001用



■ 峯 守

X-Yプロッタの製作

パルス・モータとソレノイド以外は日常品！

パーソナル・コンピュータを購入して使用していると，そのうちに周辺機器が欲しくなり，プリンタ，フロッピーディスクと次々に目が移ってしまいます．

独身貴族やブルジュワ諸氏はともかく，山の神とその手下３人に囲まれた小生などには，毎日が四面楚歌の状態です．

そこで山の神はともかく，３人の手下だけでも味方につけるべく努力している毎日です．しかし，ＣＲＴで漫画を書いても２，３回見せると，もう見向きもしないような現代っ子にサイン・ペンを使って塗り絵の原図を書いてやろうと考え，X-Yプロッタに取り組みました．

最初，BASICでパルス・モータを駆動して使用していましたが，ペンの動きが遅く，とても使いものになりませんでした．

約１箇月の使用でいたたまれず，マシン語プログラムの作成を始め，約２箇月かかってPC-8001の空きエリアにROMを実装し，N-BASICからUSR関数を使ってX-Yプロッタを動かすことに成功したので紹介します．

1 システムの構成

X-Yプロッタ製作のためにプリンタとPROMライタが追加になりました(図１)．また，PC-8001の内部にＲＯＭ基板を追加しましたが，これも自作による2716×４個のものでPC-8001のマスクＲＯＭ用のソケットに差して使用しています．

表1　I/Oポートの割り振り

I/O機器	番　地
X-Yプロッタ コントロール・ワード	70H 73H
PROMライタ コントロール・ワード	74H～76H 77H

PROM ライタは BASIC で駆動できる2716専用のものを自作しました．この基板はまたX-Yプロッタ用のI/Oポートも兼ねています．

ここで，ＰＲＯＭライタのBASICプログラムについて少し説明します．

このプログラムはＣＲＴ画面と対話型式でPROMの書き込み，読み出しが可能で，さらにイレイズ・チェックおよび，メモリとの比較が可能なものです．

参考までにプログラム・リストを**リスト１**に示します．

I/Oポートは8255を２個使用して，**表１**のように番地を割り振っています．

PROMライタの回路図は8255 1個と若干のＴＴＬ ICで簡単に製作できます．

ライト・ペンは文字データを入力するのに疲れて作りましたがフォト・トランジスタを使用したため動作が不安定でライン単位の検出しかできず，現在改造中です．

実行例

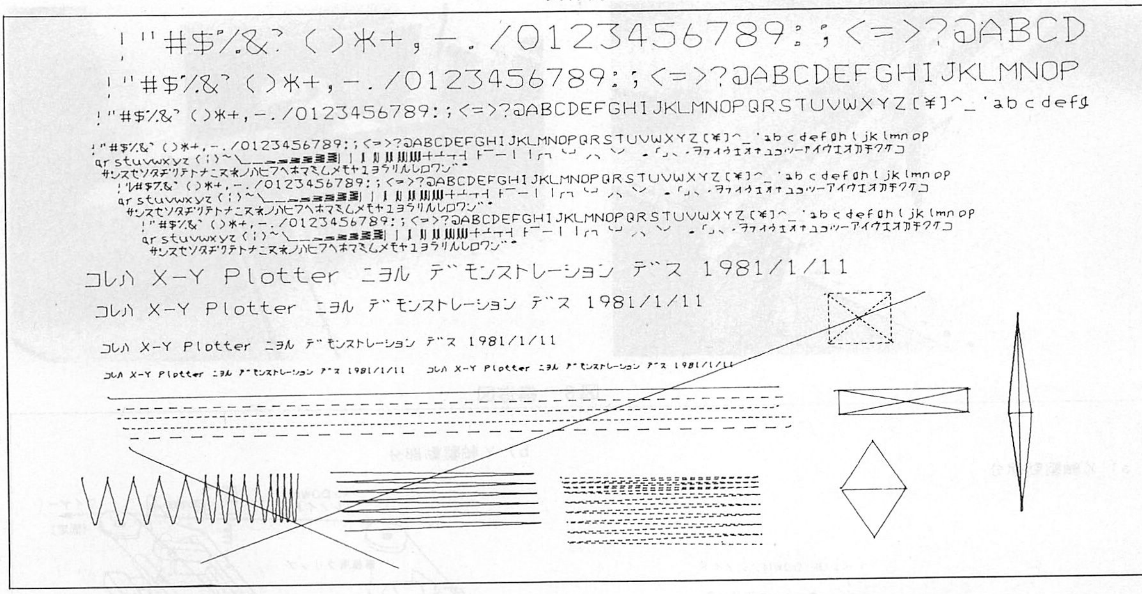


図1　システムの構成

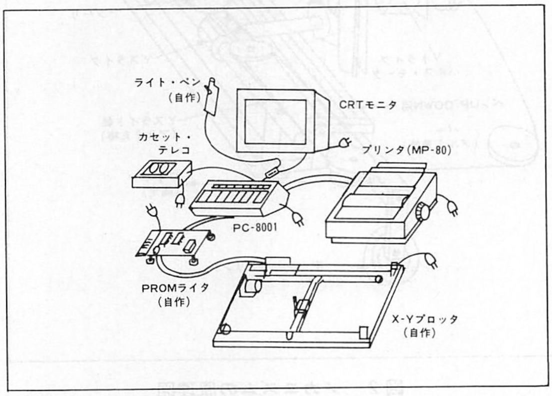


写真1　自作X-Yプロッタ

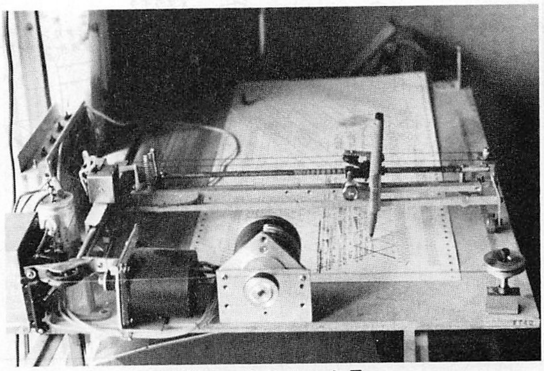
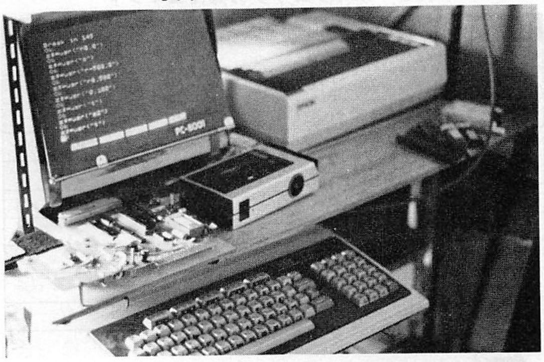


写真2　システム全景



2 X-Yプロッタの使い方

このX-Yプロッタの特長は，プロッタ自身にはパルス・モータのドライブ回路とペンUP/DOWN用ソレノイドのドライバしか持っていないことです．

コントロールするためのプログラムはPC-8001内部の空きROMエリアに実装された2716ROM２個（うち１個はキャラクタ・データ用）に書き込まれています．

X-Yプロッタの駆動はN-BASICからUSR関数を使って行ないます．

コマンドは表２のようなものがあります．

■BASICからのCALLの例

例１）Z$＝USR("P, abcdef")

この場合，ａｂｃｄｅｆと文字を書きます．

例２）Z$＝USR("m1000, 1200")

絶対座標Ｘ＝1000，Ｙ＝1200までペンUPで移動します．

例３）Z$＝USR("d0, 20, 300, 500, 100, 200, m0, 50, L5, d200, 100, h")

絶対座標（０，20）まで直線を引き，次に『(300，500)まで直線を引き，(100，200）まで直線を引く，(０，50)まで移動点線指定５ステップ間隔，(200, 100)まで直線(点線）引く原点に戻る』のようになります．

例４）Z$＝USR("T")

テスト・パターンを書きます．

例５）Z$＝USR("S5")

５倍の大きさで文字サイズを指定します．

＊　＊　＊

コマンドは大文字でも小文字でもＯＫです．

写真3　サインペンをクリップで……

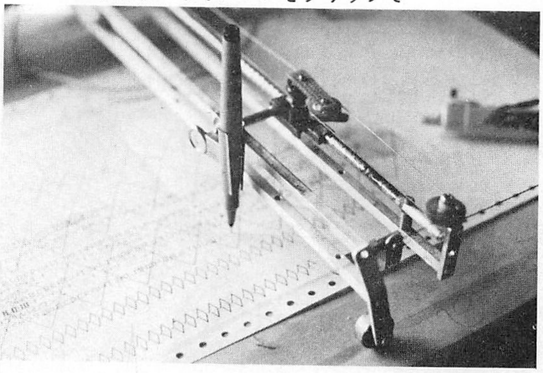


写真4　ペンUP/DOWNソレノイド部分

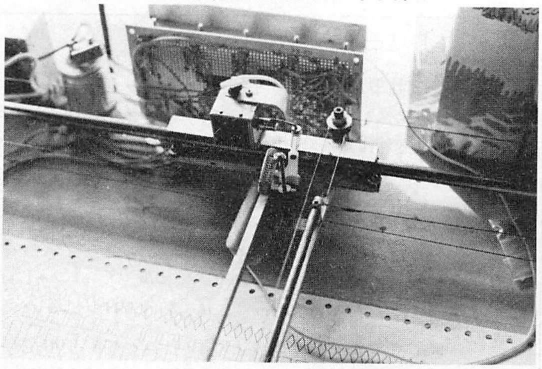


図3　構造図

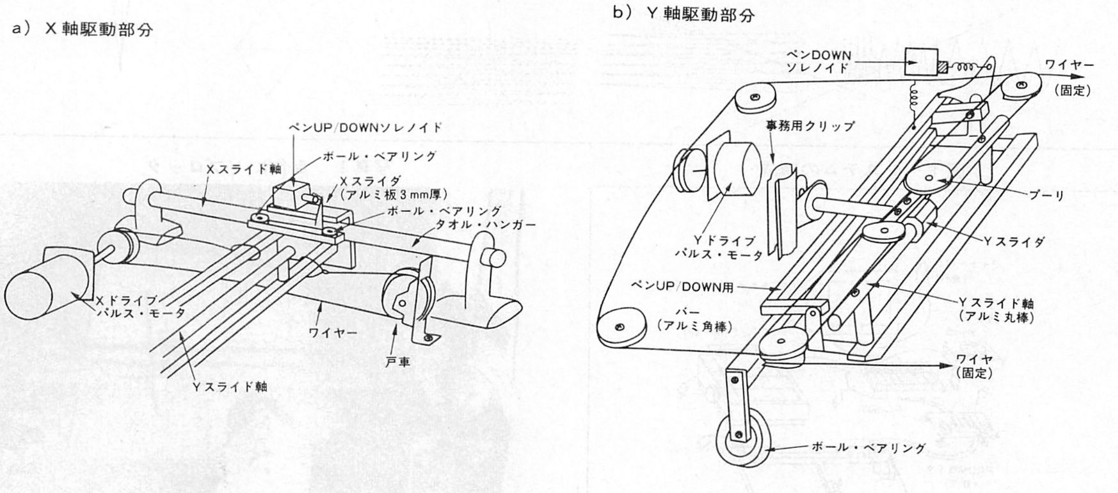
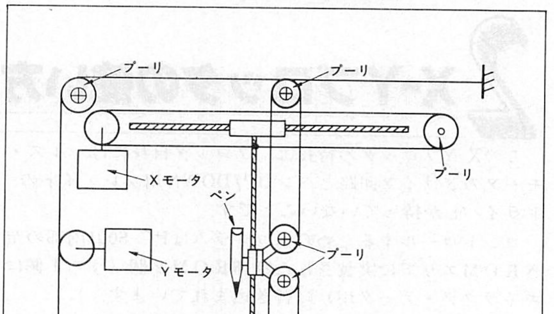


表2　X-Yプロックの操作コマンド

コマンド	意　　味
O	原点セット
H	原点にペンＵＰで戻る
M	絶対座標で示される点へペンUPで移動
R	現在位置から相対座標で示される点へペンUPで移動
D	絶対座標で示される点まで直線を引く
I	相対座標で示される点まで直線を引く
L	点線，実線と点線間隔の指定
P	文字を書く
S	文字のサイズ指定
T	テスト・パターンを書く

図2　メカニズムの原理図



3 X-Yプロッタのメカニズム

説明の前に，メカニズムといっても機械工作などとても不可能な境遇にあるため，部品はパルス・モータ２個とソレノイド以外はすべてスーパーマーケットや２年前のラジコン・ヘリの部品を流用しただけです．

X-Yプロッタの原理図および構造図を図２，３に示します．

最初はメカのガタのためまともな線を引くことはとてもできませんでした．こんなもので文字が書けるはずもないと，半ばあきらめかけていたとき，このメカニズムの欠点

写真5　パルス・モータ

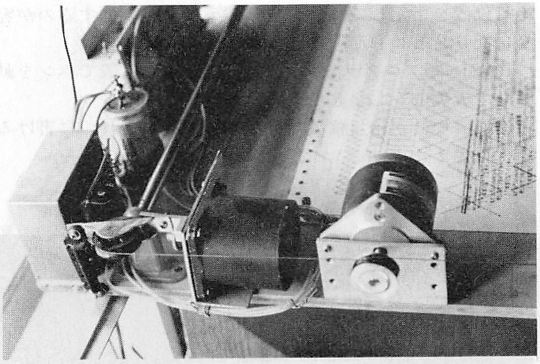


写真6　PROMライタ

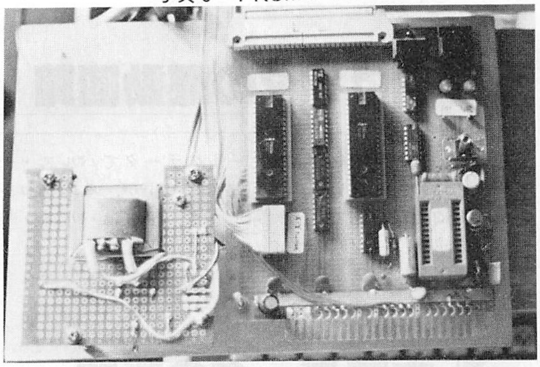


図4　ブロック図

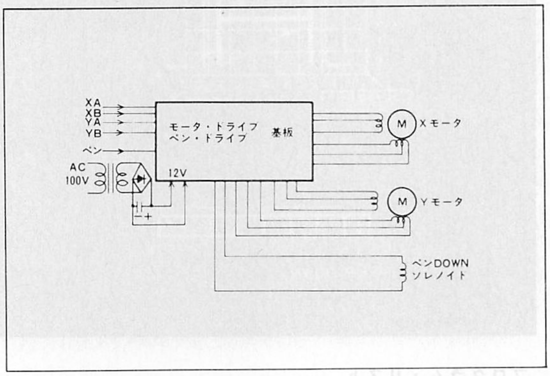


写真7　モータ・ドライブ回路

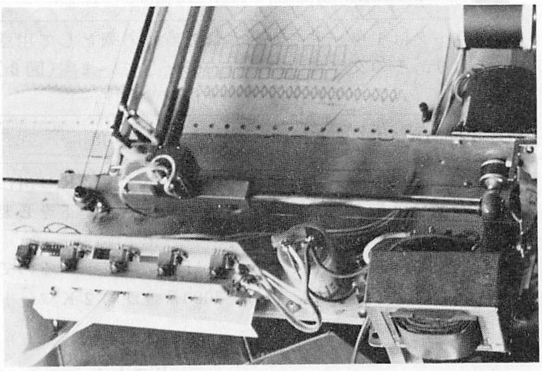


図5　モータ・ドライブ回路の原理図

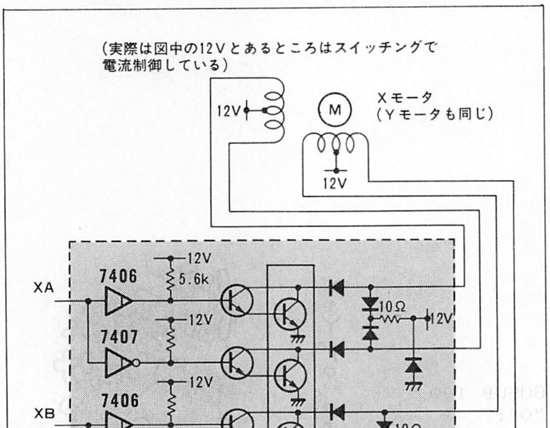


図6　モータ駆動信号のタイム・チャート

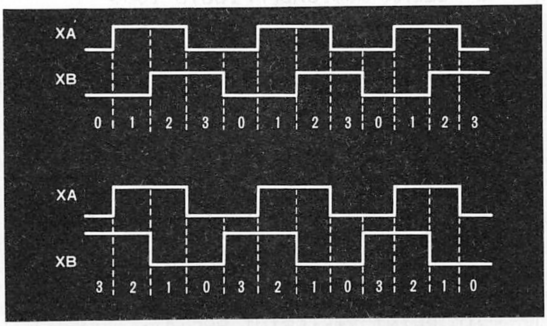
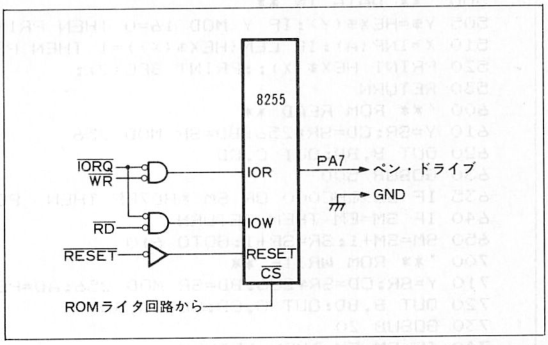


図7　X-Yプロッタ用I/Oポート



を切り抜ける糸口(まさに糸口なのです)が見つかりました。それはX-Y駆動用のワイヤーなのです。

ワイヤーだけ確実なものを使って駆動部分のまさつを極力減らすと、なんとか文字が書けたり、線が引けたりすることがわかりました。

この大切なワイヤーは釣具店に行き、石鯛釣り用のステンレス・ワイヤーを購入して使いました。

横道にそれますが、釣り具とは意外に便利なもので、使い道がたくさんあります。

たとえば、小生は投げ竿とスピニング・リールに400mの道糸を巻き、地元の富士山からゲイラカイトを上げて子供

達と楽しんでいます（本当は釣りに行けない，腹いせ）．

4 プロッタの駆動回路

プロッタの電気系統はI/Oポートのデータでパルス・モータとソレノイドを駆動する回路だけですが，そのブロック図と駆動回路を図4，5に示します．図6はパルス・モータの正回転，逆回転のタイム・チャートです．

また，図7にI/Oポート部分の回路を示します．

5 文字データの説明

文字データは1ワードで表わされる直線の列として用意されており，その文字列の終わりを00としています(図8)．

たとえば〝A″の場合，次のようになります．

```
DATA:DB 02, C2, E4, C6, 86, 22, A6, 07, 00
```

このようにしてPC-8001のキャラクタをすべてROMに書き込みましたが，データ領域として約1.6Kワードを必要とします（2716 1個弱）．

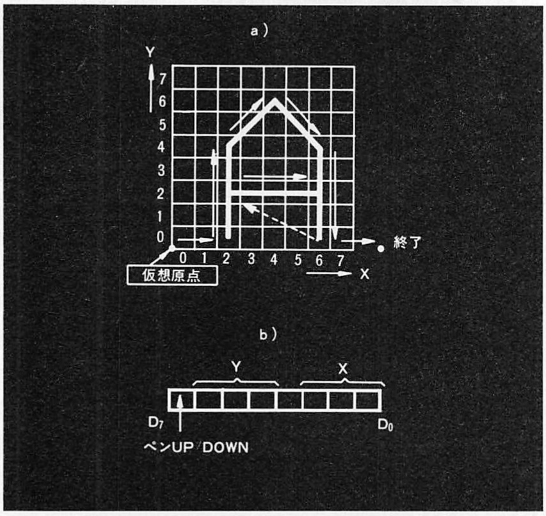
実際はこのデータに（SIZE）を掛けてプロットするので，倍率は任意に決まります．なお，7000番地から2Kワード分をデータ領域にしました．

文字を書く場合，現在ペンのある位置を仮想原点として絶対位置を指定して移動すると，データを入力するのが容易になりメモリも節約できるように思えます．

BASICでプログラミングしたとき，相対位置でペンを動かして大失敗をしました．

また，なぜか絶対位置で動かす方が字がきれいに書けるように思えます．

図8　キャラクタと文字データ



リスト1　PROMライタ　プログラム・リスト

```
5  CLEAR 300,&HBFFF:GOTO 1000
10 A=&H74:B=&H75:C=&H76:D=&H77:RETURN
20 '** WRITE PULSE **
40 OUT D,&H9:OUT D,&HB
50 LOCATE0,17:PRINT HEX$(SR):LOCATE0,19
60 OUT D,&H8:OUT D,&HA
70 RETURN
80 '** READ (ALL) **
90 OUT D,&H90
100 FOR J=0 TO 7:OUT C,J:PRINT
120 FOR I=0 TO 255:OUT B,I:Y=J*256+I
140 GOSUB 500
170 NEXT I:NEXT J:BEEP:RETURN
190 IF LEN(Y$)=1 THEN PRINT "00";
200 IF LEN(Y$)=2 THEN PRINT "0";
210 PRINT Y$;:PRINT SPC(4);
220 RETURN
500 '** DATA IN **
505 Y$=HEX$(Y):IF Y MOD 16=0 THEN PRINT : GOSUB 190
510 X=INP(A):IF LEN(HEX$(X))=1 THEN PRINT "0";
520 PRINT HEX$(X);:PRINT SPC(2);
530 RETURN
600 '** ROM READ **
610 Y=SR:CD=SR¥256:BD=SR MOD 256
620 OUT B,BD:OUT C,CD
630 GOSUB 500
635 IF SM>&HC000 OR SM<&HD7FF THEN  POKE SM,X
640 IF SM=EM THEN RETURN
650 SM=SM+1:SR=SR+1:GOTO 610
700 '** ROM WRITE **
710 Y=SR:CD=SR¥256:BD=SR MOD 256:AD=PEEK(SM)
720 OUT B,BD:OUT C,CD:OUT A,AD
730 GOSUB 20
740 IF SM=EM THEN RETURN
750 SM=SM+1:SR=SR+1:GOTO 710
800 '** ERASE CHECK **
810 Y=SR:CD=SR¥256:BD=SR MOD 256
820 OUT B,BD:OUT C,CD
```



```
830 X=INP(A) :LOCATE 2,17:PRINT HEX$(SR):LOCATE 0,19
840 IF SM=EM AND X=&HFF THEN RETURN
850 SM=SM+1:SR=SR+1
860 IF X=&HFF THEN 810
870 RETURN
900 '** Compare **
910 Y=SR:CD=SR¥256:BD=SR MOD 256:AD=PEEK(SM)
920 OUT B,BD:OUT C,CD:X=INP(A)
940 LOCATE 2,17:PRINT HEX$(SR):LOCATE 0,19
950 IF SM=EM AND AD=X THEN RETURN
960 SM=SM+1:SR=SR+1
965 IF AD<>X THEN RETURN
970 GOTO 910
1000 '**********************************
1010 '*                                *
1020 '*   P-ROM WRITER (2716)          *
1030 '*                                *
1040 '*       1980/11   BY   M.MINE    *
1050 '*                                *
1060 '**********************************
1065 WIDTH 80,25:CONSOLE 0,25,0,1:PRINT CHR$(12):GOSUB 10
1070 PRINT " ****** P-ROM Writer ******":PRINT
1080 PRINT "(1) ------ VERIFY"
1090 PRINT "(2) ------ PROGRAM"
1100 PRINT "(3) ------ ERASE CHECK"
1105 PRINT "(4) ------ COMPARE"
1110 PRINT :INPUT " Select the Number of your Task";K
1120 IF K<1 THEN 1110
1130 ON K GOSUB 2000,3000,4000,5000,6000
1140 END
1500 LINE INPUT "START ADDRESS (Memory or P-ROM)-----";SM$:GOSUB 1560
1510 LINE INPUT "E N D ADDRESS (Memory or P-ROM)-----";EM$:GOSUB 1570
1515 IF SM>=0 AND SM<&H800 THEN PRINT :SR=SM :GOTO 1590
1520 LINE INPUT "START ADDRESS (P-ROM )---------------";SR$:GOSUB 1580
1530 IF SR<0 OR SR>&H7FF THEN 1520
1550 GOTO 1590
1560 SM=VAL("&h"+SM$):RETURN
1570 EM=VAL("&h"+EM$):RETURN
1580 SR=VAL("&h"+SR$):RETURN
1590 PRINT : PRINT "SM = "+HEX$(SM),"EM = "+HEX$(EM),"SR = "+HEX$(SR)," OK ?"
1600 LINE 16,5:K$=INKEY$:IF K$=""THEN 1600 ELSE IF K$<>"y"THEN LOCATE 0,12 :GOTO
 1500
1610 RETURN
2000 '********** Verify ***********
2100 PRINT :PRINT " DATA All Print ??"
2110 K$=INKEY$:IF K$="" THEN 2110 ELSE IF K$="y" THEN GOSUB80:RETURN
2115 PRINT CHR$(30),CHR$(30),CHR$(30)
2120 PRINT :PRINT " ****** VERIFY ******":PRINT
2130 OUT D,&H90:GOSUB 1500:GOSUB600
2900 RETURN
3000 '********** Program ***********
3010 PRINT :PRINT " ****** PROGRAM ********" :PRINT
3020 OUT D,&H80:GOSUB 1500:GOSUB700
3050 RETURN
4000 '********** ERASE CHECK *********
4010 PRINT :PRINT " ****** ERASE  CHECK *****":PRINT
4015 OUT D,&H90
4020 GOSUB 1500:PRINT
4030 GOSUB 800
4040 IF X=&HFF THEN BEEP:RETURN
4045 PRINT HEX$(SR-1),HEX$(X)
4050 FOR I=0 TO 10:BEEP1:FOR J=0 TO 10:NEXTJ:BEEP0:FOR J=0 TO 10:NEXTJ:NEXTI
4060 K$=INKEY$:IF K$="" THEN 4060
4070 GOTO 4030
5000 '********** COMPARE ****************
5010 PRINT :PRINT " ****** COMPARE *******":PRINT
5020 OUT D,&H90
5030 GOSUB 1500:PRINT
5035 GOSUB900
5040 IF SM=EM THEN BEEP:RETURN
5050 PRINT HEX$(SR-1),HEX$(X)
5060 FOR I=0 TO 10:BEEP1:FOR J=0 TO 10:NEXTJ:BEEP0:FOR J=0 TO 10:NEXTJ:NEXTI
5070 K$=INKEY$:IF K$="" THEN 5070
5080 GOTO 5035
```

```
                        ; **********************
                        ; *                    *
                        ; *  X-Y Plotter V1.0  *
                        ; *                    *
                        ; *     1981/1         *
                        ; *                    *
                        ; *      By M.MINE     *
                        ; *                    *
                        ; **********************
                        ;
F2C0                    ABX:   EQU  0F2C0H  ①
F2C2                    ABY:   EQU  0F2C2H  ②
F2C4                    FLX:   EQU  0F2C4H
F2C6                    FLY:   EQU  0F2C6H
F2C8                    MDX:   EQU  0F2C8H  ③
F2C9                    MDY:   EQU  0F2C9H  ④
F2CA                    ABMX:  EQU  0F2CAH
F2CC                    ABMY:  EQU  0F2CCH
F2D0                    MDB:   EQU  0F2D0H  ⑤
F2D1                    LDR:   EQU  0F2D1H  ⑥
F2D3                    PNTTM: EQU  0F2D3H  ⑦
F2D4                    FXYR:  EQU  0F2D4H  ⑧
F2E0                    PFG1:  EQU  0F2E0H  ⑨
F2E1                    PNTR:  EQU  0F2E1H  ⑩
F2E2                    PFG2:  EQU  0F2E2H  ⑪
0070                    PORT:  EQU  70H     ⑫
0600                    XMAX:  EQU  600H
04C0                    YMAX:  EQU  4C0H

F2CE                    CHBF:  EQU  0F2CEH
F2CF                    SIZE:  EQU  0F2CFH
F2D5                    IPX:   EQU  0F2D5H
F2D7                    IPY:   EQU  0F2D7H

F2E3                    MODER: EQU  0F2E3H
F2E4                    MDNR:  EQU  0F2E4H
EF59                    STRDR: EQU  0EF59H
44C7                    ASKTB: EQU  44C7H
7000                    DATA:  EQU  7000H
                        ;
                        ;
                               ORG  6800H
***** USR(X-Y PLOTTER) *****
                        ;
6800 C3626B                    JP   XYPLTR
                        ;
                        ;
                        ;Initialize
                        ;
6803 AF                 INT:   XOR  A
6804 67                        LD   H,A
6805 6F                        LD   L,A
6806 22C0F2                    LD   (ABX),HL  ⑬
6809 22C2F2                    LD   (ABY),HL
680C 32E2F2                    LD   (PFG2),A
680F 32D4F2                    LD   (FXYR),A
6812 32E1F2                    LD   (PNTR),A
6815 32D3F2                    LD   (PNTTM),A
6818 3E80                      LD   A,80H
681A D373                      OUT  (73H),A
681C C9                        RET



                        ;
681D 78                 MDT:   DB   78H       ⑭
                        ;
681E 21C8F2             XP1:   LD   HL,MDX
6821 7E                        LD   A,(HL)
6822 3C                        INC  A
6823 E603                      AND  03H
6825 77                        LD   (HL),A
6826 2AC0F2                    LD   HL,(ABX)
6829 23                        INC  HL
682A 22C0F2                    LD   (ABX),HL
682D C9                        RET
                        ;
682E 21C8F2             XM1:   LD   HL,MDX
6831 7E                        LD   A,(HL)
6832 3D                        DEC  A
6833 E603                      AND  03H
6835 77                        LD   (HL),A
6836 2AC0F2                    LD   HL,(ABX)
6839 2B                        DEC  HL
683A 22C0F2                    LD   (ABX),HL
683D C9                        RET
                        ;
                        ;
683E 21C9F2             YP1:   LD   HL,MDY
6841 7E                        LD   A,(HL)
6842 3C                        INC  A
6843 E603                      AND  03H
6845 77                        LD   (HL),A
6846 2AC2F2                    LD   HL,(ABY)
6849 23                        INC  HL
684A 22C2F2                    LD   (ABY),HL
684D C9                        RET
                        ;
684E 21C9F2             YM1:   LD   HL,MDY
6851 7E                        LD   A,(HL)
6852 3D                        DEC  A
6853 E603                      AND  03H
6855 77                        LD   (HL),A
6856 2AC2F2                    LD   HL,(ABY)
6859 2B                        DEC  HL
685A 22C2F2                    LD   (ABY),HL
685D C9                        RET
                        ;
685E 3AC8F2             MBS:   LD   A,(MDX)   ⑮
6861 47                        LD   B,A
6862 CD7768                    CALL DCNV
6865 32D0F2                    LD   (MDB),A
6868 3AC9F2                    LD   A,(MDY)
686B 47                        LD   B,A
686C CD7768                    CALL DCNV
686F 07                        RLCA
6870 07                        RLCA
6871 21D0F2                    LD   HL,MDB
6874 B6                        OR   (HL)
6875 77                        LD   (HL),A
6876 C9                        RET
                        ;
6877 3A1D68             DCNV:  LD   A,(MDT)   ⑯
687A 4F                        LD   C,A
687B 78                        LD   A,B


687C A7                        AND  A
687D CA8868             DCNV1: JP   Z,DCNV2
6880 79                        LD   A,C
6881 0F                        RRCA
6882 0F                        RRCA
6883 4F                        LD   C,A
6884 05                        DEC  B
6885 C37D68                    JP   DCNV1
                        ;
6888 79                 DCNV2: LD   A,C
6889 E603                      AND  03H
688B C9                        RET
                        ;
                        ;Pen up/down Control
                        ;SET MDB
688C 21D0F2             PUDS:  LD   HL,MDB    ⑰
688F 3AE0F2                    LD   A,(PFG1)
6892 CB5F                      BIT  3,A
6894 CA9A68                    JP   Z,PUDS1
6897 CBFE                      SET  7,(HL)
6899 C9                        RET
689A CBBE               PUDS1: RES  7,(HL)
689C C9                        RET
                        ;
                        ;Motor output with delay
                        ;
689D 3AE2F2             POUT:  LD   A,(PFG2)
68A0 21E0F2                    LD   HL,PFG1
68A3 BE                        CP   (HL)
68A4 C2B068                    JP   NZ,POUT1
68A7 3AD0F2             POUT2: LD   A,(MDB)
68AA D370                      OUT  (PORT),A
68AC CDCD68                    CALL DLYM
68AF C9                        RET
68B0 3AD0F2             POUT1: LD   A,(MDB)
68B3 D370                      OUT  (PORT),A
68B5 CDBF68                    CALL DLYP
68B8 7E                        LD   A,(HL)
68B9 32E2F2                    LD   (PFG2),A
68BC C3A768                    JP   POUT2
                        ;
68BF D5                 DLYP:  PUSH DE        ⑱
68C0 110030                    LD   DE,3000H
68C3 1D                 DLY1:  DEC  E
68C4 C2C368                    JP   NZ,$-1
68C7 15                        DEC  D
68C8 C2C368                    JP   NZ,$-5
68CB D1                        POP  DE
68CC C9                        RET
                        ;
68CD D5                 DLYM:  PUSH DE        ⑲
68CE 110006                    LD   DE,600H
68D1 C3C368                    JP   DLY1
                        ;
                        ;
************ ﾁｮｸｾﾝ ﾎｶﾝ ***************
                        ;
68D4 F5                 XD1:   PUSH AF        ⑳
68D5 D5                        PUSH DE
68D6 E5                        PUSH HL
68D7 CD196A                    CALL MXCX



68DA DAEE68                    JP   C,XYD12
68DD 21C5F2                    LD   HL,FLX+1
68E0 CB7E                      BIT  7,(HL)
68E2 C2EB68                    JP   NZ,XD11
68E5 CD1E68                    CALL XP1
68E8 C3EE68                    JP   XYD12
68EB CD2E68             XD11:  CALL XM1
68EE CD1269             XYD12: CALL PNT       ㉑
68F1 E1                        POP  HL
68F2 D1                        POP  DE
68F3 F1                        POP  AF
68F4 C9                        RET
                        ;
68F5 F5                 YD1:   PUSH AF        ㉒
68F6 D5                        PUSH DE
68F7 E5                        PUSH HL
68F8 CD276A                    CALL MXCY
68FB DAEE68                    JP   C,XYD12
68FE 21C7F2                    LD   HL,FLY+1
6901 CB7E                      BIT  7,(HL)
6903 C20C69                    JP   NZ,YD11
6906 CD3E68                    CALL YP1
6909 C3EE68                    JP   XYD12
690C CD4E68             YD11:  CALL YM1
690F C3EE68                    JP   XYD12
                        ;
6912 21E0F2             PNT:   LD   HL,PFG1
6915 CB7E                      BIT  7,(HL)
6917 C8                        RET  Z
6918 3AE1F2                    LD   A,(PNTR)
691B FE00                      CP   0H
691D CA3569                    JP   Z,PNT1
6920 21D3F2                    LD   HL,PNTTM
6923 BE                        CP   (HL)
6924 F5                        PUSH AF
6925 34                        INC  (HL)
6926 F1                        POP  AF
6927 C0                        RET  NZ
6928 AF                        XOR  A
6929 77                        LD   (HL),A
692A 21E0F2                    LD   HL,PFG1
692D CB5E                      BIT  3,(HL)
692F CA3569                    JP   Z,PNT1
6932 CB9E                      RES  3,(HL)
6934 C9                        RET
6935 CBDE               PNT1:  SET  3,(HL)
6937 C9                        RET
                        ;
                        ;Write Straight Line
                        ;
6938 F5                 WSTL:  PUSH AF
6939 C5                        PUSH BC
693A D5                        PUSH DE
693B E5                        PUSH HL

693C 21C5F2                    LD   HL,FLX+1  ㉓
693F CB7E                      BIT  7,(HL)
6941 C29269                    JP   NZ,ABBC
6944 ED4BC4F2                  LD   BC,(FLX)
6948 21C7F2             ADS:   LD   HL,FLY+1  ㉔



694B CB7E                      BIT  7,(HL)
694D C29E69                    JP   NZ,ABDE
6950 ED5BC6F2                  LD   DE,(FLY)

6954 210000             DCK:   LD   HL,0H
6957 ED4A                      ADC  HL,BC     ㉕
6959 C26C69                    JP   NZ,DCK1
695C 210000                    LD   HL,0H
695F ED5A                      ADC  HL,DE     ㉖
6961 C27E69                    JP   NZ,DCK3
6964 CD096A                    CALL DRV
6967 E1                 RET:   POP  HL
6968 D1                        POP  DE
6969 C1                        POP  BC
696A F1                        POP  AF
696B C9                        RET
                        ;
696C 210000             DCK1:  LD   HL,0H
696F ED5A                      ADC  HL,DE     ㉗
6971 C28869                    JP   NZ,DCK2
6974 ED43D1F2           DCK4:  LD   (LDR),BC  ㉘
6978 CDB469                    CALL XLTY
697B C36769                    JP   RET
697E ED53D1F2           DCK3:  LD   (LDR),DE  ㉙
6982 CDAA69                    CALL XSTY
6985 C36769                    JP   RET
                        ;
6988 6B                 DCK2:  LD   L,E
6989 62                        LD   H,D
698A ED42                      SBC  HL,BC     ㉚
698C D27E69                    JP   NC,DCK3
698F C37469                    JP   DCK4
                        ;
6992 2B                 ABBC:  DEC  HL
6993 7E                        LD   A,(HL)
6994 2F                        CPL
6995 4F                        LD   C,A
6996 23                        INC  HL
6997 7E                        LD   A,(HL)
6998 2F                        CPL
6999 47                        LD   B,A
699A 03                        INC  BC
699B C34869                    JP   ADS
                        ;
699E 2B                 ABDE:  DEC  HL
699F 7E                        LD   A,(HL)
69A0 2F                        CPL
69A1 5F                        LD   E,A
69A2 23                        INC  HL
69A3 7E                        LD   A,(HL)
69A4 2F                        CPL
69A5 57                        LD   D,A
69A6 13                        INC  DE
69A7 C35469                    JP   DCK
                        ;
69AA EB                 XSTY:  EX   DE,HL     ㉛
69AB 59                        LD   E,C
69AC 50                        LD   D,B
69AD 4D                        LD   C,L
69AE 44                        LD   B,H
69AF 3E01                      LD   A,1H




69B1 32D4F2                    LD   (FXYR),A
69B4 60                 XLTY:  LD   H,B
69B5 69                        LD   L,C       ㉜
69B6 CB3D                      SRL  L
69B8 CB3C                      SRL  H         ㉝
69BA AF                 XLT1:  XOR  A         ㉞
69BB ED52                      SBC  HL,DE     ㉟
69BD CAF369                    JP   Z,XLT2
69C0 DAF369                    JP   C,XLT2
69C3 3AD4F2                    LD   A,(FXYR)
69C6 FE00                      CP   0H
69C8 C2006A                    JP   NZ,XLT6
69CB CDD468                    CALL XD1       ㊱
69CE CD096A                    CALL DRV
69D1 E5                 XLT4:  PUSH HL
69D2 2AD1F2                    LD   HL,(LDR)
69D5 2B                        DEC  HL
69D6 22D1F2                    LD   (LDR),HL
69D9 7D                        LD   A,L
69DA FE00                      CP   0H
69DC CAE369                    JP   Z,XLT3
69DF E1                        POP  HL
69E0 C3BA69                    JP   XLT1
                        ;
69E3 7C                 XLT3:  LD   A,H
69E4 FE00                      CP   0H
69E6 CAED69                    JP   Z,XLT5
69E9 E1                        POP  HL
69EA C3BA69                    JP   XLT1
                        ;
69ED E1                 XLT5:  POP  HL
69EE AF                        XOR  A
69EF 32D4F2                    LD   (FXYR),A
69F2 C9                        RET
```

```
                ;
69F3 09         XLT2:  ADD  HL,BC
69F4 CDD468            CALL XD1
69F7 CDF568            CALL YD1
69FA CD096A            CALL DRV
69FD C3D169            JP   XLT4
                ;
6A00 CDF568     XLT6:  CALL YD1
6A03 CD096A            CALL DRV
6A06 C3D169            JP   XLT4

*** Motor & Pen Drive ******
                ;
6A09 F5         DRV:   PUSH AF          (37)
6A0A C5                PUSH BC
6A0B D5                PUSH DE
6A0C E5                PUSH HL
6A0D CD5E68            CALL MBS         (38)
6A10 CD8C68            CALL PUDS        (39)
6A13 CD9D68            CALL POUT        (40)
6A16 C36769            JP   RET
                ;
                ; LIMIT CHECK
                ;
6A19 210006     MXCX:  LD   HL,XMAX     (41)
6A1C ED5BC0F2          LD   DE,(ABX)
6A20 AF                XOR  A
6A21 ED52              SBC  HL,DE
6A23 D8                RET  C
6A24 7A                LD   A,D
6A25 07                RLCA
6A26 C9                RET
6A27 21C004     MXCY:  LD   HL,YMAX     (42)
6A2A ED5BC2F2          LD   DE,(ABY)
6A2E AF                XOR  A
6A2F ED52              SBC  HL,DE
6A31 D8                RET  C
6A32 7A                LD   A,D
6A33 07                RLCA
6A34 C9                RET

***** MOVE ABSOLUTELY *****

6A35 F5         ABDRW: PUSH AF
6A36 C5                PUSH BC
6A37 D5                PUSH DE
6A38 E5                PUSH HL
6A39 2ACAF2            LD   HL,(ABMX)
6A3C ED5BC0F2          LD   DE,(ABX)
6A40 AF                XOR  A
6A41 ED52              SBC  HL,DE
6A43 22C4F2            LD   (FLX),HL
6A46 2ACCF2            LD   HL,(ABMY)
6A49 ED5BC2F2          LD   DE,(ABY)
6A4D AF                XOR  A
6A4E ED52              SBC  HL,DE
6A50 22C6F2            LD   (FLY),HL
6A53 CD3869            CALL WSTL
6A56 C36769            JP   RET
                ;
***** HOME *****
                ;
6A59 F5         HOME:  PUSH AF
6A5A E5                PUSH HL
6A5B AF                XOR  A
6A5C 210000            LD   HL,0H
6A5F 22CAF2            LD   (ABMX),HL
6A62 22CCF2            LD   (ABMY),HL
6A65 32E0F2            LD   (PFG1),A
6A68 CD356A            CALL ABDRW
6A6B E1                POP  HL
6A6C F1                POP  AF
6A6D C9                RET
                ;
***** IMAGINARY ORIGIN SET *****
                ;
6A6E E5         ISET:  PUSH HL
6A6F 2AC0F2            LD   HL,(ABX)
6A72 22D5F2            LD   (IPX),HL
6A75 2AC2F2            LD   HL,(ABY)
6A78 22D7F2            LD   (IPY),HL
6A7B E1                POP  HL
6A7C C9                RET

***** CHARACTER DRAW *****
                ;
6A7D F5         CDRW:  PUSH AF
6A7E C5                PUSH BC
6A7F D5                PUSH DE
6A80 E5                PUSH HL
6A81 32CEF2            LD   (CHBF),A
6A84 3E00              LD   A,0
6A86 010000            LD   BC,0
6A89 32E0F2            LD   (PFG1),A
6A8C 32E1F2            LD   (PNTR),A
6A8F 3ACEF2            LD   A,(CHBF)
6A92 CB7F              BIT  7,A
6A94 CA9C6A            JP   Z,CDRW1     (43)
6A97 3EFF              LD   A,0FFH      (44)
6A99 32E0F2            LD   (PFG1),A
6A9C 3ACEF2     CDRW1: LD   A,(CHBF)
6A9F CDC06A            CALL CDRW3
6AA2 2AD5F2            LD   HL,(IPX)
6AA5 19                ADD  HL,DE
6AA6 22CAF2            LD   (ABMX),HL
6AA9 3ACEF2            LD   A,(CHBF)
6AAC 0F                RRCA
6AAD 0F                RRCA
6AAE 0F                RRCA
6AAF 0F                RRCA
6AB0 CDC06A            CALL CDRW3
6AB3 2AD7F2            LD   HL,(IPY)
6AB6 19                ADD  HL,DE
6AB7 22CCF2            LD   (ABMY),HL
6ABA CD356A            CALL ABDRW
6ABD C36769            JP   RET
                ;
6AC0 E607       CDRW3: AND  7           (45)
6AC2 1600              LD   D,0
6AC4 5F                LD   E,A
6AC5 3ACFF2            LD   A,(SIZE)
6AC8 4F                LD   C,A
6AC9 7B                LD   A,E
6ACA 83                ADD  A,E
6ACB 0D                DEC  C
6ACC C2CA6A            JP   NZ,$-2
6ACF 5F                LD   E,A
6AD0 C9                RET
                ;
***** Plot 1 Character (A=ASKII) *****
                ;
6AD1 F5         PLTC:  PUSH AF
6AD2 C5                PUSH BC
6AD3 D5                PUSH DE
6AD4 E5                PUSH HL
6AD5 D620              SUB  20H
6AD7 DAE76A            JP   C,PLT1
6ADA 4F                LD   C,A
6ADB 210070     PLT2:  LD   HL,DATA
6ADE 7E         PLT3:  LD   A,(HL)
6ADF A7                AND  A
6AE0 CAEC6A            JP   Z,PLT4
6AE3 23                INC  HL
6AE4 C3DE6A            JP   PLT3
                ;
6AE7 0E20       PLT1:  LD   C,20H
6AE9 C3DB6A            JP   PLT2
                ;
6AEC 79         PLT4:  LD   A,C
6AED A7                AND  A
6AEE 23                INC  HL
6AEF CAF66A            JP   Z,WCHR
6AF2 0D                DEC  C
6AF3 C3DE6A            JP   PLT3
                ;
6AF6 CD6E6A     WCHR:  CALL ISET        (46)
6AF9 7E                LD   A,(HL)
6AFA A7                AND  A
6AFB CA6769            JP   Z,RET
6AFE CD7D6A            CALL CDRW
6B01 23                INC  HL
6B02 C3F96A            JP   $-9
                ;
6B05 3E20       TEST:  LD   A,20H
6B07 0670              LD   B,70H
6B09 CDD16A     TEST1: CALL PLTC
6B0C B8                CP   B
6B0D D2146B            JP   NC,TEST3
6B10 3C                INC  A
6B11 C3096B            JP   TEST1
6B14 CD3C6B     TEST3: CALL CRLF
6B17 3E71              LD   A,71H
6B19 06BA              LD   B,0BAH
6B1B CDD16A     TEST4: CALL PLTC
6B1E B8                CP   B
6B1F D2266B            JP   NC,TEST5
6B22 3C                INC  A
6B23 C31B6B            JP   TEST4
6B26 CD3C6B     TEST5: CALL CRLF
6B29 3EBB              LD   A,0BBH
6B2B 06DF              LD   B,0DFH
6B2D CDD16A     TEST6: CALL PLTC
6B30 B8                CP   B
6B31 D2386B            JP   NC,TEST2
6B34 3C                INC  A
6B35 C32D6B            JP   TEST6
6B38 CD3C6B     TEST2: CALL CRLF
6B3B C9                RET
                ;
6B3C F5         CRLF:  PUSH AF
6B3D C5                PUSH BC
6B3E D5                PUSH DE
6B3F E5                PUSH HL
6B40 210000            LD   HL,0
6B43 22CAF2            LD   (ABMX),HL
6B46 3E07              LD   A,7
6B48 CDC06A            CALL CDRW3
6B4B ED52              SBC  HL,DE
6B4D 2ACCF2            LD   HL,(ABMY)
6B50 AF                XOR  A
6B51 ED52              SBC  HL,DE
6B53 110500            LD   DE,5
6B56 AF                XOR  A
6B57 ED52              SBC  HL,DE
6B59 22CCF2            LD   (ABMY), HL
6B5C CD356A            CALL ABDRW
6B5F C36769            JP   RET

***** X-Y PLOTTER *****
                ;
6B62 F5         XYPLTR:PUSH AF
6B63 C5                PUSH BC
6B64 D5                PUSH DE
6B65 E5                PUSH HL
6B66 2A59EF            LD   HL,(STRDR)
6B69 010000            LD   BC,0
6B6C 110000            LD   DE,0
6B6F FE03              CP   3
6B71 C2D26C            JP   NZ,VARSET
6B74 7E         MODS:  LD   A,(HL)
6B75 FE20              CP   20H
6B77 CA7F6B            JP   Z,$+8       (47)
6B7A FE2C              CP   2CH
6B7C C2836B            JP   NZ,$+7      (48)
6B7F 23                INC  HL
6B80 C3746B            JP   MODS
6B83 FE2D              CP   2DH         (49)
6B85 CAE26B            JP   Z,NSET
6B88 FE30              CP   30H
6B8A DA6769            JP   C,RET       (50)
6B8D FE3A              CP   3AH
6B8F DAE26B            JP   C,NSET      (51)
6B92 FE41              CP   41H
6B94 DA6769            JP   C,RET       (52)
6B97 CBAF              RES  5,A         (53)
6B99 FE5A              CP   5AH
6B9B D26769            JP   NC,RET      (54)
6B9E FE44              CP   44H         (55)
6BA0 CA576C            JP   Z,DRW       (56)
6BA3 FE48              CP   48H         (57)
6BA5 CA4E6C            JP   Z,ROP       (58)
6BA8 FE49              CP   49H         (59)
6BAA CA696C            JP   Z,RDRW      (60)
6BAD FE4C              CP   4CH         (61)
6BAF CA0A6C            JP   Z,LINE      (62)
6BB2 FE4D              CP   4DH         (63)
6BB4 CA6E6C            JP   Z,MOVE      (64)
6BB7 FE4E              CP   4EH         (65)
6BB9 CA166C            JP   Z,MARK      (66)
6BBC FE4F              CP   4FH         (67)
6BBE CA3C6C            JP   Z,ORGS      (68)
6BC1 FE50              CP   50H         (69)
6BC3 CA196C            JP   Z,PRINT     (70)
6BC6 FE51              CP   51H         (71)
6BC8 CA2A6C            JP   Z,ALPROT    (72)
6BCB FE52              CP   52H         (73)
6BCD CA7E6C            JP   Z,RMOVE     (74)
6BD0 FE53              CP   53H         (75)
6BD2 CA2D6C            JP   Z,SCLE      (76)
6BD5 FE54              CP   54H         (77)
6BD7 CA456C            JP   Z,TST       (78)
6BDA FE58              CP   58H         (79)
6BDC CA396C            JP   Z,AXIS      (80)
6BDF C36769            JP   RET
                ;
6BE2 FE2D       NSET:  CP   2DH
6BE4 CAFD6B            JP   Z,NSET1
6BE7 2B                DEC  HL
6BE8 CDC744            CALL ASKTB       (81)
6BEB CD836C     NSET2: CALL DTST        (82)
6BEE 7E                LD   A,(HL)
6BEF FE30              CP   30H
6BF1 DA746B            JP   C,MODS
6BF4 FE3A              CP   3AH
6BF6 D2746B            JP   NC,MODS
6BF9 23                INC  HL
6BFA C3EE6B            JP   $-12
                ;
6BFD CDC744     NSET1: CALL ASKTB
6C00 7B                LD   A,E
6C01 2F                CPL
6C02 5F                LD   E,A
6C03 7A                LD   A,D
6C04 2F                CPL
6C05 57                LD   D,A
6C06 13                INC  DE
6C07 C3EB6B            JP   NSET2

***** LINE TYPE & DOT *****
6C0A CDC744     LINE:  CALL ASKTB       (83)
6C0D 3E0F              LD   A,0FH
6C0F A3                AND  E
6C10 32E1F2            LD   (PNTR),A
6C13 C3746B            JP   MODS
                ;
***** Write MARK *****
                ;
6C16 C3746B     MARK:  JP   MODS

***** Write CHARACTER *****
                ;
6C19 06A0       PRINT: LD   B,0A0H
6C1B 23                INC  HL
6C1C 23                INC  HL
6C1D 7E         PRINT1:LD   A,(HL)
6C1E FE22              CP   22H         (84)
6C20 CA746B            JP   Z,MODS
6C23 CDD16A            CALL PLTC        (85)
6C26 23                INC  HL
6C27 C31D6C            JP   PRINT1
                ;
***** ALPHA ROTATE *****
                ;
6C2A C3746B     ALPROT:JP   MODS
                ;
***** ALPHA SCALE *****
                ;
6C2D CDC744     SCLE:  CALL ASKTB       (86)
6C30 3E07              LD   A,7
6C32 A3                AND  E
6C33 32CFF2            LD   (SIZE),A
6C36 C3746B            JP   MODS

***** AXIS PLOT *****
                ;
6C39 C3746B     AXIS:  JP   MODS
                ;
***** ORIGIN SET *****
                ;
6C3C E5         ORGS:  PUSH HL
6C3D CD0368            CALL INT
6C40 E1                POP  HL

6C41 23.               INC  HL
```

```
6C42 C3746B         JP    MODS
                ;
***** TEST PATERN *****
                ;
6C45 E5       TST:  PUSH  HL
6C46 CD056B         CALL  TEST
6C49 E1             POP   HL
6C4A 23             INC   HL
6C4B C3746B         JP    MODS

***** Return to ORIGINAL POINT *****
                ;
6C4E E5       ROP:  PUSH  HL
6C4F CD596A         CALL  HOME
6C52 E1             POP   HL
6C53 23             INC   HL
6C54 C3746B         JP    MODS

***** DRW *****
                ;
6C57 3E44     DRW:  LD    A,44H
6C59 32E3F2   DRW1: LD    (MODER),A
6C5C 3EFF           LD    A,0FFH
6C5E 32E0F2         LD    (PFG1),A    ⑧⑦
6C61 AF             XOR   A
6C62 32E4F2         LD    (MDNR),A
6C65 23             INC   HL
6C66 C3746B         JP    MODS
                ;
***** RELATIVE DRAW *****
                ;
6C69 3E49     RDRW: LD    A,49H
6C6B C3596C         JP    DRW1

***** MOVE *****
                ;
6C6E 3E4D     MOVE: LD    A,4DH
6C70 32E3F2   MOVE1: LD   (MODER),A
6C73 AF             XOR   A
6C74 32E0F2         LD    (PFG1),A    ⑧⑧
6C77 32E4F2         LD    (MDNR),A
6C7A 23             INC   HL
6C7B C3746B         JP    MODS

***** RELATIVE MOVE *****
                ;
6C7E 3E52     RMOVE: LD   A,52H
6C80 C3706C         JP    MOVE1

***** DATA SET *****

6C83 E5       DTST: PUSH  HL
6C84 3AE3F2         LD    A,(MODER)
6C87 FE44           CP    44H         ⑧⑨
6C89 CA9E6C         JP    Z,DSTA
6C8C FE4D           CP    4DH         ⑨⓪
6C8E CA9E6C         JP    Z,DSTA
6C91 FE49           CP    49H         ⑨①
6C93 CAB86C         JP    Z,DSTF
6C96 FE52           CP    52H         ⑨②

6C98 CAB86C         JP    Z,DSTF
6C9B E1             POP   HL
6C9C 23             INC   HL
6C9D C9             RET
                ;
6C9E 21E4F2   DSTA: LD    HL,MDNR     ⑨③
6CA1 7E             LD    A,(HL)
6CA2 A7             AND   A
6CA3 C2AD6C         JP    NZ,DSTA1    ⑨④
6CA6 ED53CAF2       LD    (ABMX),DE
6CAA 34             INC   (HL)
6CAB E1             POP   HL
6CAC C9             RET
                ;
6CAD ED53CCF2 DSTA1: LD   (ABMY),DE
6CB1 AF             XOR   A
6CB2 77             LD    (HL),A
6CB3 CD356A         CALL  ABDRW
6CB6 E1             POP   HL
6CB7 C9             RET
                ;
6CB8 21E4F2   DSTF: LD    HL,MDNR     ⑨⑤
6CBB 7E             LD    A,(HL)
6CBC A7             AND   A
6CBD C2C76C         JP    NZ,DSTF1
6CC0 ED53C4F2       LD    (FLX),DE
6CC4 34             INC   (HL)
6CC5 E1             POP   HL
6CC6 C9             RET
                ;
6CC7 ED53C6F2 DSTF1: LD   (FLY),DE
6CCB AF             XOR   A
6CCC 77             LD    (HL),A
6CCD CD3869         CALL  WSTL
6CD0 E1             POP   HL
6CD1 C9             RET

***** VAR. SET *****
                ;
6CD2 C3746B   VARSET:JP   MODS        ⑨⑥
```







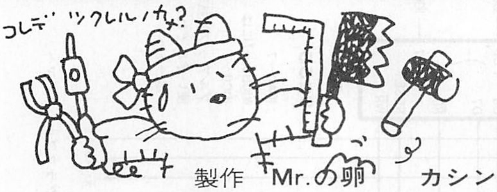
プログラムの注釈

① ;Absolute pos.X
② ;   Y
③ ;Motor drive reg.X
④ ;   Y
⑤ ;Motor drive port Buffer
⑥ ;Step Length
⑦ ;Point Timing Reg.
⑧ ;|dX|>|dY| THEN 0
⑨ ;Pen flag1 00..up FF..down
⑩ ;Point distance 00 to 7F
⑪ ;Pen flag2
⑫ ;Output port
⑬ ;Origin SET
⑭ ;TABLE for Motor output
⑮ ;Motor Buffer Set by MDX,MDY
⑯ ;Data conversion
⑰ ;if PFG1<>0 then PEN down
⑱ ;Pen up/down Delay
⑲ ;Motor output Delay
⑳ ;1step move X direction
㉑ ;Point Distance
㉒ ;1step move Y direction
㉓ ;Absolute dX set BC
㉔ ;Absolute dY set DE
㉕ ;dX=0 ?
㉖ ;dY=0 ?
㉗ ;dY=0 ?
㉘ ;|dX|>|dY|
㉙ ;|dY|>=|dX|
㉚ ;if |dX|>|dY| then Set Carry
㉛ ;|dX|<|dY|
㉜ ;|dX|>|dY|
㉝ ;D=dX/2 D=dY/2
㉞ ;Carry=0
㉟ ;D=D-dY D=D-dX
㊱ ;1 Step for X
㊲ ;Motor & Pen OUTPUT
㊳ ;Motor Buffer Set
㊴ ;pen up down Set
㊵ ;port out
㊶ ;MAX.check X
㊷ ;MAX.check Y
㊸ ;Pen up
㊹ ;Pen down
㊺ ;multiply by SIZE
㊻ ;Write 1 Character
㊼ ;A‡'SPACE'
㊽ ;A‡','
㊾ ;'-'
㊿ ;A<'0'
51 ;Data Set
52 ;Mode<'A'
53 ;A=a
54 ;Mode>'Z'
55 ;D
56 ;DRAW
57 ;H
58 ;RETURN TO ORIGINAL POIN
59 ;I
60 ;RELATIVE DRAW
61 ;L
62 ;LINE TYPE&DOT
63 ;M
64 ;MOVE
65 ;N
66 ;Write MARK
67 ;O
68 ;ORIGIN SET
69 ;P
70 ;Write CHARACTER
71 ;Q
72 ;ALPHA ROTATE
73 ;R
74 ;RELATIVE MOVE
75 ;S
76 ;ALPHA SCALE
77 ;T
78 ;TEST PATERN
79 ;X
80 ;AXIS PLOT
81 ;ASKII to BINARY
82 ;DATA SET
83 ;DATA Read
84 ;A='"'
85 ;PLOT CHARACTER
86 ;Data Read
87 ;PEN Down
88 ;PEN Up
89 ;DRAW
90 ;MOVE
91 ;RDRW
92 ;RMOVE
93 ;ABS MOVE DATA SET
94 ;SECOND DATA
95 ;FLOAT. MOVE
96 ;Dummy

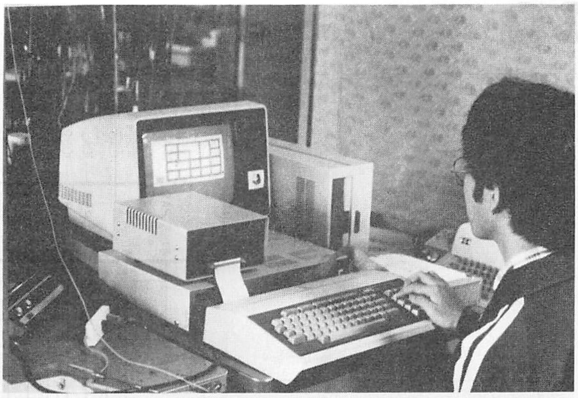
ラベル・テーブル

ABBC	6992	ABDE	699E	ABDRW	6A35	ABMX	F2CA	ABMY	F2CC
ABX	F2C0	ABY	F2C2	ADS	6948	ALPROT	6C2A	ASKTB	44C7
AXIS	6C39	CDRW	6A7D	CDRW1	6A9C	CDRW3	6AC0	CHBF	F2CE
CRLF	6B3C	DATA	7000	DCK	6954	DCK1	696C	DCK2	6988
DCK3	697E	DCK4	6974	DCNV	6877	DCNV1	687D	DCNV2	6888
DLY1	68C3	DLYM	68CD	DLYP	68BF	DRV	6A09	DRW	6C57
DRW1	6C59	DSTA	6C9E	DSTA1	6CAD	DSTF	6CB8	DSTF1	6CC7
DTST	6C83	FLX	F2C4	FLY	F2C6	FXYR	F2D4	HOME	6A59
INT	6803	IPX	F2D5	IPY	F2D7	ISET	6A6E	LDR	F2D1
LINE	6C0A	MARK	6C16	MBS	685E	MDB	F2D0	MDNR	F2E4
MDT	681D	MDX	F2C8	MDY	F2C9	MODER	F2E3	MODS	6B74
MOVE	6C6E	MOVE1	6C70	MXCX	6A19	MXCY	6A27	NSET	6BE2
NSET1	6BFD	NSET2	6BEB	ORGS	6C3C	PFG1	F2E0	PFG2	F2E2
PLT1	6AE7	PLT2	6ADB	PLT3	6ADE	PLT4	6AEC	PLTC	6AD1
PNT	6912	PNT1	6935	PNTR	F2E1	PNTTM	F2D3	PORT	0070
POUT	689D	POUT1	68B0	POUT2	68A7	PRINT	6C19	PRINT1	6C1D
PUDS	688C	PUDS1	689A	RDRW	6C69	RET	6967	RMOVE	6C7E
ROP	6C4E	SCLE	6C2D	SIZE	F2CF	STRDR	EF59	TEST	6B05
TEST1	6B09	TEST2	6B38	TEST3	6B14	TEST4	6B1B	TEST5	6B26
TEST6	6B2D	TST	6C45	VARSET	6CD2	WCHR	6AF6	WSTL	6938
XD1	68D4	XD11	68EB	XLT1	69BA	XLT2	69F3	XLT3	69E3
XLT4	69D1	XLT5	69ED	XLT6	6A00	XLTY	69B4	XM1	682E
XMAX	0600	XP1	681E	XSTY	69AA	XYD12	68EE	XYPLTR	6B62
YD1	68F5	YD11	690C	YM1	684E	YMAX	04C0	YP1	683E

PC-8001用



製作　Mr.の卵　カシン
設計とポート解読　Dr.の卵　Yoh
ポート解読とご指導　Prof.の卵　P.



汎用インターフェイスの製作

やっと入手した憧れのPC-8001でゲームを楽しむのにも少し飽き，さて，シンセサイザでもつなごうか，それとも実験のお守りでもさせようか，はたまた，ロボットの脳みそにでもしようかなどと考えたあなた．目の前に，ハードウェア非公開の厚い壁が見えませんか．

しょうがないこととあきらめて，大枚はたいて，拡張ユニットPC-8011を買っても，アマチュアに使いでのある汎用並列ポートは，入力12ビット，出力12ビットだけです．

そこで！ここにご紹介するは入出力しめて7バイト，おまけにタイマを持つI/Oポートです．我々は使っちゃいけない本体のバスを使っていますが，そこはアマチュア，恐れていては…．

それと，付録に『NEC完全非保証　PC本体I/Oポート』を載せました．眉に唾をつけながらご利用ください．

写真1　インターフェイス・ユニットのリアパネル

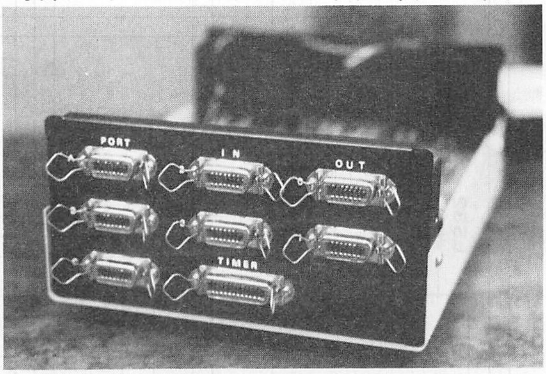


表1　メモリ・マップ

10進	16進	用　途	接続されているIC	備　考
128	80	出　力(0)	8212	ストローブ付き
129	81	出　力(1)	8212	ストローブ付き
130	82	入　力(2)	LS244	——
131	83	入　力(3)	LS244	——
132	84	入出力(A)	8255(ポートA)	——
133	85	入出力(B)	8255(ポートB)	——
134	86	入出力(C)	8255(ポートC)	——
135	87	入出力コントロール	8255	——
136	88	タイマ0	8253	1kHz
137	89	タイマ1	8253	1kHz
138	8A	タイマ2	8253	4kHz
139	8B	タイマ・コントロール	8253	

1. 回路概説，etc.

図1の回路図がほとんどのことを語ってくれています．この汎用I/Oポートは以下の機能を持っています．

❶**汎用入力ポート**　2バイト　LS244
❷**汎用出力ポート**　2バイト　8212×2
❸**汎用入出力ポート**　3バイト　8255A×1
❹**汎用タイマ**　3チャンネル　8253×1

私達は，これを汎用基板ヒノマル-3の上に，ラッピングを使って組み立てました．すでに2台が完成し，現在，3台目の製作途中です．充分再現性があると考えています．表1に示すように，このI/Oポートは＄80～＄8BのI/Oアドレスを占めています．これはPC-8011やディスク，あるいは本体のアドレスと競合しないように定めたものです．

本体と，この装置の間は，山一のFDS-50-03(02も可，本体側）とFAP-50-03（装置側），および当然ながら50芯のフラット・ケーブルで接続されています．ケーブルの長さは私の場合20cmとしましたが，これはできるだけ短くしてください．短ければ短い方がよいのです．

お金がたくさんあって，PC-8011拡張ユニットも持ち，さらにこのI/Oポートも製作する方は，どうぞここの接続はご自分でbang bangしてください．きっと，何か適当なコネクタがあるでしょうから…．

また，大阪ICMのCP/Mアダプタを買った方は，亜土電子で取り扱っている信号取り出し用コネクタ，AP PROD UCTSのイントラ・コネクタというのが使えます．

私達はケースに鈴蘭堂のMEC3Bを，また，入出力用コネクタには第一電子の57型というものを使いました．この辺はまったく趣味の問題です．

2. 作り終わったら

まず，配線をチェックします．チェックが終わったら，もう1度チェックします．何せ，使ってはならないシステム・バスを使わせていただくのです．このことの恐れ多さにおののき，さらにもう1度配線をチェックします．

この慎重さがあなたのPC-8001を守るのです．もう1度やりますか．それもいいですね．

図1(a) 全回路図

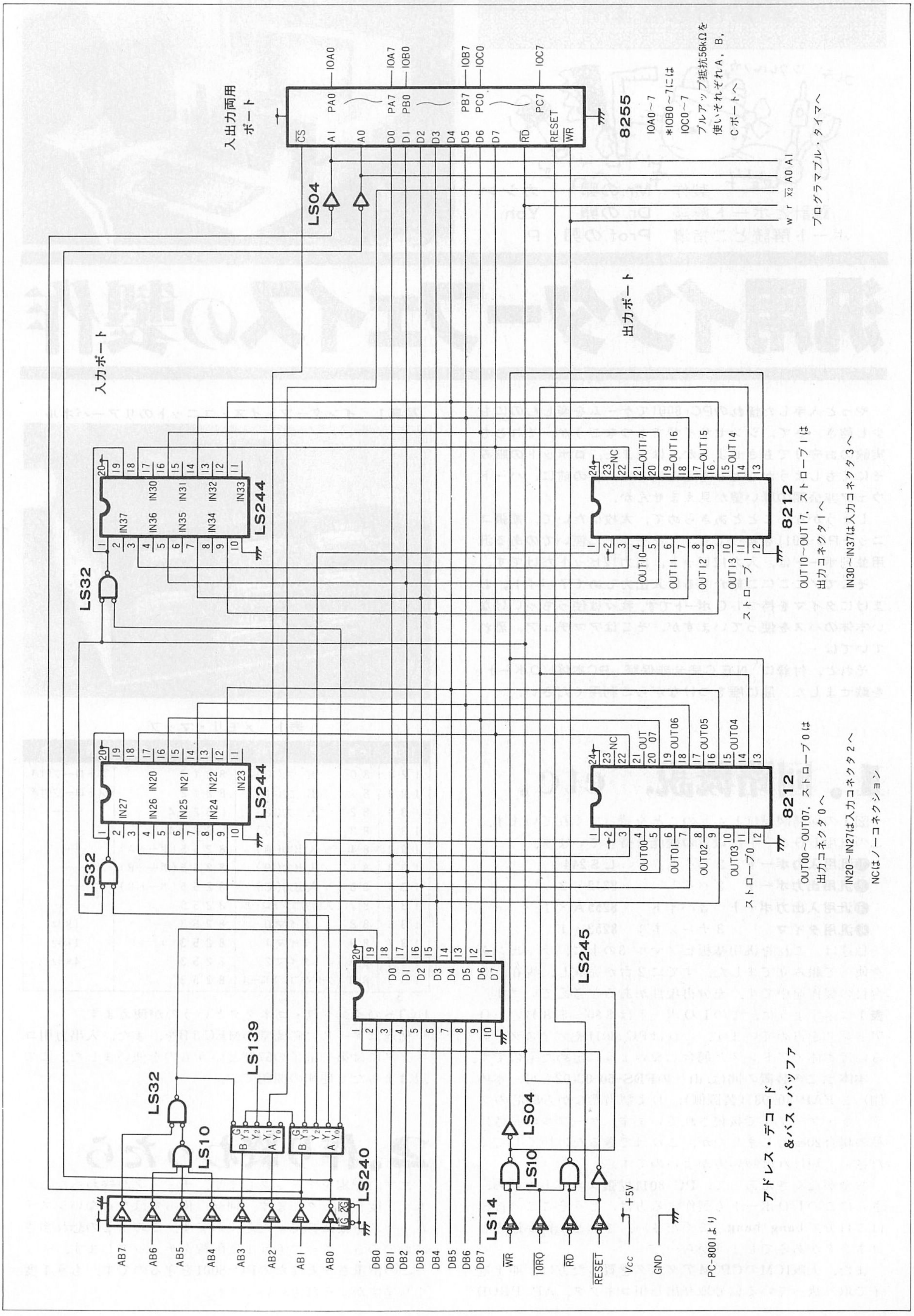

8255
入出力両用ポート
入力ポート
出力ポート
IOA0〜7 *IOB0〜7には
IOC0〜7 プルアップ抵抗56kΩを使いそれぞれA, B, Cポートへ
プログラマブル・タイマへ
LS04
LS244
LS32
8212
ストローブ1
ストローブ0
OUT10〜OUT17, ストローブ1は出力コネクタ1へ
IN30〜IN37は入力コネクタ3へ
OUT00〜OUT07, ストローブ0は出力コネクタ0へ
IN20〜IN27は入力コネクタ2へ
NCはノーコネクション
LS245
LS139
LS10
LS240
LS14
アドレス・デコード
&バス・バッファ
PC-8001より
AB7
AB6
AB5
AB4
AB3
AB2
AB1
AB0
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
WR
IORQ
RD
RESET
Vcc
GND
+5V

図1(b)

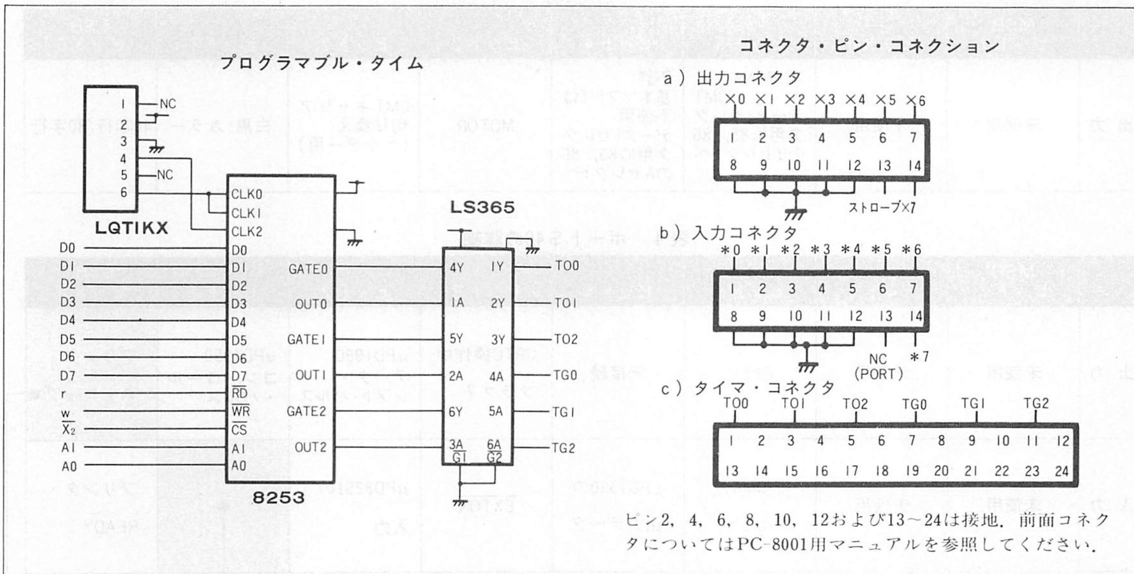


写真2　ユニット内部の様子（ＩＣ間の配線はラッピングを使用）

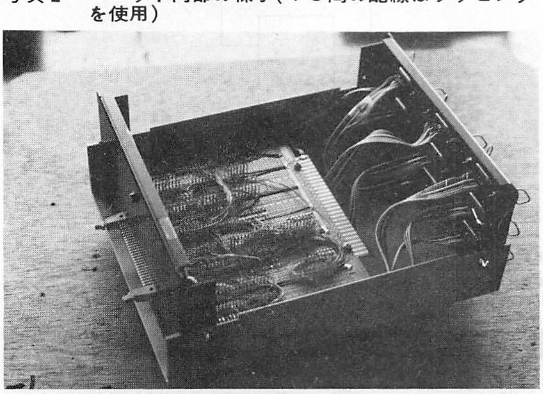
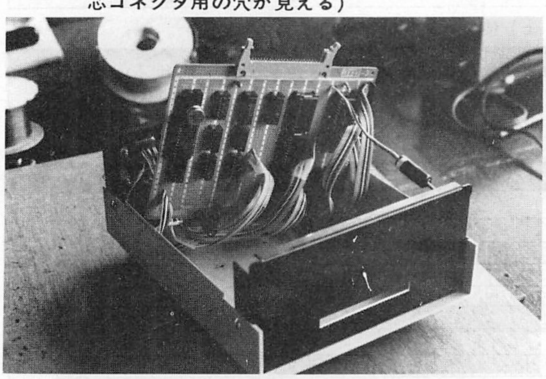


写真3　基板を起こしたところ（フロント・パネルには50芯コネクタ用の穴が見える）



★CP/Mアダプタ，拡張ユニットを持っている方へ！

大阪ＩＣＭのCP/MアダプタやＰＣの拡張ユニットＰＣ8041を持っている方は，バスの競合が生じないよう，このI/Oポートを70H～にリロケートする必要があります．それにはアドレス・デコードの上位4bitを図のようにする必要があります（動作確認済）．

ただし，ＩＣＭのＣＰ/Ｍアダプタ使用者は，そのI/O用バスから信号を取れば，このままの回路でも可です．

また，今秋発売予定のＰＣ-8042と組み合わせるにはそれ用の基板にこのままの回路で組み上げて，マザーボードにさせばおそらく大丈夫のはずです．

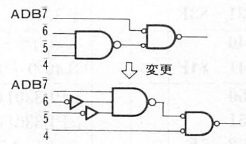
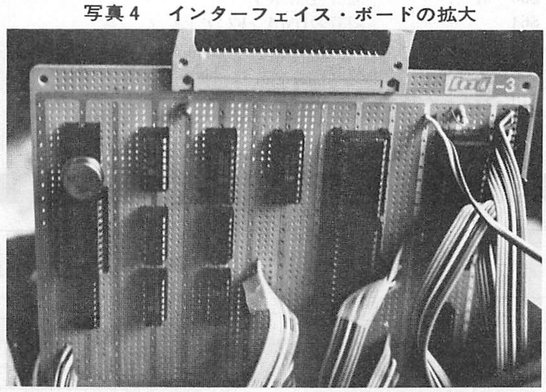


写真4　インターフェイス・ボードの拡大



3．使用上の注意

その1　私達は電源を本体からイタダキました．約0.5Aになります．本体の通風，放熱には充分注意してください．もちろん，電源を内蔵してもいいわけですが，このときは本体の電源を必ず先に入れ，切るときはこのI/Oポートから切るようにします．

その2　上に関連して，TTLは必ずLSタイプを使ってください．もっとも8212などという大喰いを使ってますが，これは足元にころがっていたからで他意はありません．

その3　電源を入れたままで，PC-8001と本機の接続を行なったり，あるいは外したりは，$\lim_{n\to\infty}(\dot{絶}\dot{対})^n$（ただし，絶対＞1）しないでください．思わぬ事故のもとです．我々は使ってはならないものを使っているのです．

表3　ポート＄30の詳細

	MSB ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	LSB ビット0
出力	未使用	未使用	TERMINAL /CMT データ・セレクタ用IC85, 86のBセレクトへ	不詳 基本ソフトでは不使用 データ・セレクタ用IC85, 86のAセレクトへ	MOTOR	CMTキャリア切り換え（ヘッダー用）	白黒/カラー	40字行/80字行

表4　ポート＄40の詳細

	MSB ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	LSB ビット0
出力	未使用	未使用	BEEP	未接続	CRTC操作中フラグ？	μPD1990 データ・シフト・パルス	μPD1990 コントロール・パルス	プリンタ・ストローブ
入力	未使用	未使用	DMA BUSY	μPD1990の出力データ	EXTON	μPD8251の入力		プリンタ・READY

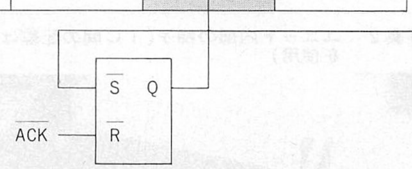


表2　PC-8001 I/Oポート・マップ（＄0～＄7F）

アドレス	入出力 入	入出力 出	ポートの内容
$0～$9	○		キーボード・センス線
$A～$F			未使用
$10		○	プリンタ用およびμPD1990用データ
$11～$1F			＄10のイメージ
$20	○	○	8251（UART）データ・ワード
$21	○	○	8251（UART）コントロール・ワード
$22～$2F			＄20，21のイメージ
$30		○	並列出力ポート（IC27）表3参照
$31～$3F			＄30のイメージ
$40	○	○	入出力ポート（入力IC25，出力IC26）表4参照
$41～$4F			＄40のイメージ
$50		○	μPD3301（CRTC）データ・ワード表5参照
$51		○	μPD3301（CRTC）コントロール・ワード
$52～5F			＄50，＄51のイメージ
$60～6F			8257（DMAC）
$64		○	8257のDMA ch2 アドレス
$65		○	8257の ch2 ターミナル・カウント
$68		○	8257のモードセット・レジスタ
$70～$7F			未使用

表5　間違いだらけのハズ（？）のμPD3301の使い方

❶コントロール・ワード＄51を送る．
❷その後，必要個数のデータ（＄50）を書き込む．

b）使用法2　カーソル設定

手順	設定データ
1	$51 ← コマンド（＄81）
2	$50 ← データ1（Xの値）
3	$50 ← データ2（Yの値）

c）使用法3　カーソル消去

手順	消去データ
1	$51 ← コマンド（＄80）

a）使用法1　イニシャライズ・データ

手順	インシャライズ・データ	
1	$51 ←コマンド（$00）	
2	$50 ← データ1	ビット7：？／ビット6～1：（1行のドット数/16）－1／ビット0：インタレース/ノンインタレース
3	$50 ← データ2	ビット7～6：ブリンク用分周数／ビット5～0：表示行数－1
4	$50 ← データ3	ビット7：？／ビット6：カーソルEN／ビット5：カーソル・ブリンク／ビット4～0：1行のラスタ数－1
5	$50 ← データ4	ビット7～5：垂直同期パルス幅／ビット4：垂直同期EN？／ビット3～1：水平同期パルス幅／ビット0：SCANに関する何か，カラー時のみ？
6	$50 ← データ5	ビット7：アトリ方式フラグ？／ビット6：カラー/モノ／ビット5：アトリビュート有効フラグ？／ビット4～1：（アトリビュート・エリア/2）－1
7	$51 ← コマンド2（＄43，ただし＄41でも可か？）	
8	$51 ← コマンド3	ビット7：？／ビット6：？／ビット5：表示EN／ビット4：？／ビット3：？／ビット2：？／ビット1：？／ビット0：反転

その4　出力ポート，あるいは入出力ポートを出力として使うときは，他のICの出力と接続したり，電源あるいはアースに直接つながらないようにしてください．

また，入出力ポートを入力として外部に接続した場合，誤ってこのポートを出力にプログラムすると，上に述べたようなことが期せずして起こってしまいます．これを未然に防ぐため，図のように入力として使う入出力ポートには1kΩ程度の抵抗を接続しておくと安全です．

4.PC-8001本体のI/Oマップ

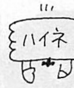
本体に割り当ててあるアドレスは＄0～＄7Fです．これをまず＄10ごとにデコードし，さらに必要な場合は下位4ビットのデコードも行なっているようです．ただし，その下位4ビットのデコードを行なっていないところも多くあり，その場合，たとえば＄31～＄3Fは＄30のイメージとなるわけです．

表2はアドレス・マップです．アドレス，および，そのポートを入出力のどちらに使っているか，そして，何かつながっているかを示しています．＄30，＄40，そしてCRTCについては，さらに詳しく，解読結果を示しました．

ただし，これらの情報をうのみにして，何かをしたために起きたことは，すべてあなたの責任であると思ってご利用ください．

μPD1990，8251，8257などについては，データが公開されています．これらのポートの詳しいことについては，各ICのデータ・シートを参考にしてください．

5.おわりに



何も教えてもらえないといってダダをこねるよりも，自ら『勝ち取る』方針が大事だと思います．ただし，ここにご紹介した汎用I/Oポートにせよ，本体のI/Oマップにせよ，思わぬミスはいくらでもあるかもしれません．

ここに示したのは，たたき台，あるいは皆さんのより良きマイコン・ライフへの踏み台のつもりなのです．皆さんの責任の上において，この記事が利用され，何かのお役に立てれば，うれしい限りです．

これ以上のことがわかったら，ぜひI/Oに発表してください．そのようなフィード・バックこそが大事なのではないでしょうか．

最後に一言．PC-8001を真の国民機とするため，速やかな**『ハードの公開を！』**希望しています．

●ありがとう

椚田鷹男様，あなたの作られた逆アセンブラこそ使いはしませんでしたけど，あなたが調べられたことは，大変役に立ちましたので，ここに記して感謝の意を表します．

付録　アトリビュート・エリアの内容の意味

アトリビュート・エリアの先頭↓

偶数アドレス	奇数アドレス	偶数アドレス	奇数アドレス

偶数アドレスの内容	その桁から，次の奇数アドレスの内容に従ったアトリビュートを行なう．
奇数アドレスの内容	アトリビュートの内容を表わす．

a）白黒モード時

ビット	7	6	5	4	3	2	1	0
モード	M	—	D	P	—	R	B	S

b）カラーモード時(bit3=0か1かで異なる)

ビット	7	6	5	4	3	2	1	0
モード	—	—	D	P	0	R	B	S
	カラー			M	1	—	—	—

M：グラフィック/文字モード　R：リバース
D：アンダーライン　B：ブリンク（S指定の時無効）
P：アッパーライン　S：シークレット

簡単に作れてローコスト！

PC-8001 直結

PROM WRITERの製作

長岡　美紀雄

最近の2716の値下りはものすごく，2708と同じくらいの値段まで下りました．そこでロー・コストで製作が容易な2716PROMライタを作りました．

ハードウェア

使用パーツからテストまで

主となるインターフェイスLSIは，NECのμPD8255AC-5でモード０で使用しています．

その他，I/Oアドレスのデコードに74LS138，コントロール・バスのデコードに74LS32，V_{pp}コントロールにオープン・コレクタ タイプの74LS03を使用しています．

このようにLSI 1個，TTL３個の簡単な構成で PROMライタができるのですから，皆さんもぜひ作ってみてください．

■回路について

μPD8255AC-5はモード０で使い，$PA_{0\sim7}$を2716のデータ・バス$D_{0\sim7}$に接続し，書き込みのとき，出力ポート，イレーズ・チェック，ベリファイ，ロードのとき入力ポートと切り換えて使っています(図１)．

$PB_{0\sim7}$はアドレス下位用で出力ポートとして使用し，$A_{0\sim7}$に接続します．ＰＣはアドレス上位用とともに各制御信号を出力します．

$PC_{0\sim2}$を$A_{8\sim10}$に，PC_5は$\overline{CE}$/PGM制御に，PC_6をV_{pp}制御に，PC_7を$\overline{OE}$制御にそれぞれ使用しています．

この場合，アドレスに必要な各制御信号をＯＲ・ＡＮＤして，ON/OFFしています．

74LS138はアドレス・デコーダで，PC-8001内部で使用していないアドレス80H～FFHまでを10Hおきにデコードします（Y_0～Y_7は，80H～F0H）．

8255のI/OアドレスはＰＡが80H，ＰＢが81H，ＰＣが82H，CW（コントロール・ワード)が83Hです．

74ＬＳ32はコントロール・バスのデコードで，PC-8001(Z80)の$\overline{IORQ}$，$\overline{WR}$，$\overline{RD}$の各信号から$\overline{IORD}$，$\overline{IOWR}$を作り8255の$\overline{RD}$，$\overline{WR}$に加えています．

74ＬＳ03はオープン・コレクタ出力で，ピン配置は74ＬＳ00と同じです．このゲートはPC-8001からの$\overline{RESET}$信号を反転して8255に加えたり，V_{pp}コントロールに使用したり，$\overline{OE}$のコントロールにも使用しています．

写真１　PROMライタの本体

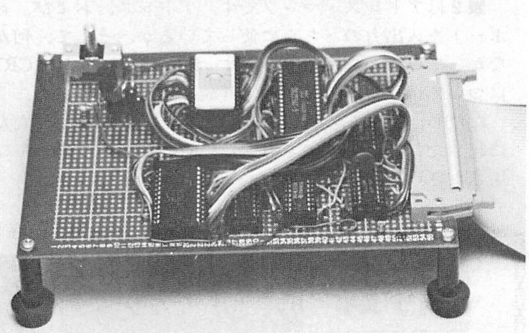
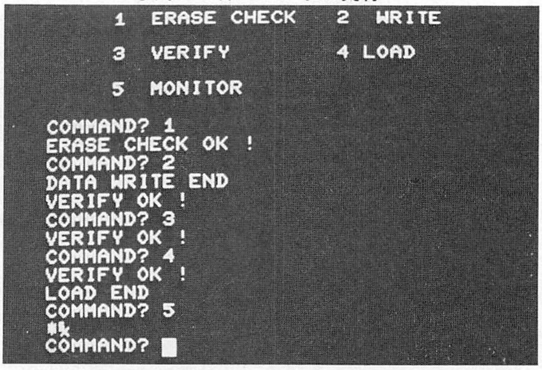


写真２　各コマンドの実行



回路図中＋５Ｖ，ＳＷ５Ｖ，ＳＷ28Ｖというのがありますが，これは，ＳＷが付くものはＲＯＭを抜き差しするときにON/OFFするスイッチでON/OFFされる電圧です．

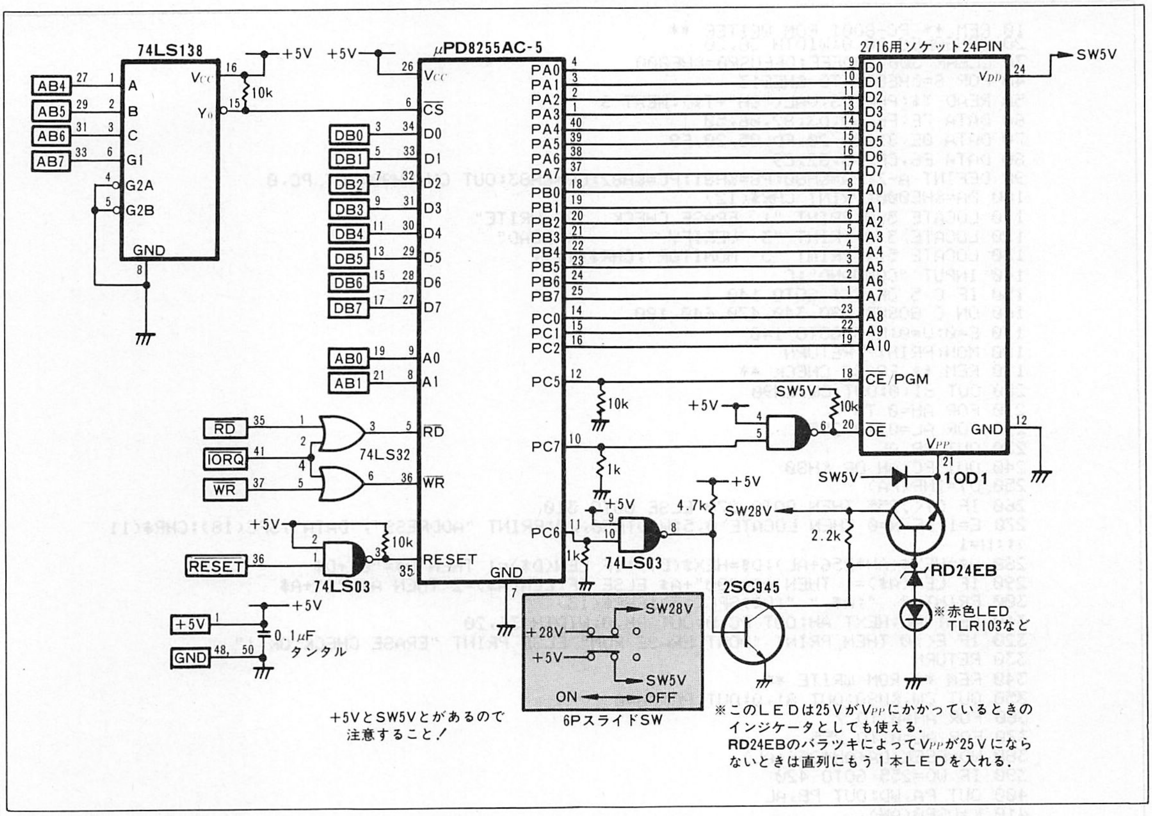
これは，ＲＯＭを抜き差しするとき，2716に電圧が加わっていてはまずいので，前記の$\overline{OE}$などのプルアップ抵抗にも加えています．

■PC-8001との接続

PC-8001とは，50芯フラット・ケーブルで接続します．ＰＣ側カードエッジ・コネクタに山一のFDS-50-13B，反対側にFAS-50-03を使用します．

そして，プリント板にFAP-50-03＃2を使ってＰＣから

図1　PROMライタ全回路図



の信号を受けます．フラット・ケーブルは短かい方が良いのですが，私の場合，1mでも誤動作しないで動いています．

■製作上の注意

フリント基板はICピッチのICB-96を使い，LSI，ICはすべてソケットを使いました．

PC-8001からの信号はすべて出力（データ・バスは，入力のときもある）なので誤配線をしたり，ショートしたりするとそのままPCが〝サービス・センター行き〟となるので，配線が終わったら，自分が納得いくまで確認してください．

■テスト

まず初めに，ICを差さずにPCと本機を接続し電源を入れます．そして各ソケットの V_{cc} とGND間に＋5Vが加わっているかどうか調べます．

OKなら1度電源を切り，ICをソケットに差し込みます．再び電源を入れ，ダイレクト・モードで

```
OUT & H83, & H80
```

を実行します．

そして2716ソケットの A_0～A_{10}，D_0～D_7，$\overline{CE}$/PGM端子が0V，V_{pp}，$\overline{OE}$がSWをONにしたとき5V，OFFのとき0V(本当は，オープン)であれば，まずはOKです．続いて

```
OUT &H80, 255, OUT &H81, 255, OUT &H82, 255
```

を実行します．そして A_0～A_{10}，D_0～D_7，$\overline{CE}$/PGMがHレベル（3～5V），$\overline{OE}$が0V，V_{pp}がSW ONで25V，OFFで0VならOKです．

V_{pp}は，RD24EBのバラツキによって25Vにならないときがあります．そのとき24V以上であればそのままでOKですが，そうでないときは，赤色LEDをもう1本直列に追加してください．

また，このLEDは V_{pp} に25Vが加っているときのインジケータとしても利用できます．

以上がOKならPCのリセット・ボタンを押し，リスト1を入力します．

ソフトウェア
使い方から使用感

■使い方

プログラムをRUNすると，コマンド・リストを表示し，

```
COMMAND？
```

と聞いてくるので，必要なコマンドを入力してください．次に各コマンドの説明をします．

1．イレーズ・チェック

ROM内のデータがすべてFFHかどうか，チェックします．1つでもFFHでなければそのアドレスとデータを表示し，

```
DONT ERASE ROM
```

リスト1　PROMライタ・プログラム

```
10 REM ** PC-8001 ROM WRITER **
20 CONSOLE6,20,0:WIDTH 36,20
30 CLEAR 300,&HDFFF:DEFUSR0=&HE800
40 FOR S=&HE800 TO &HE813
50 READ T$:POKE S,VAL("&H"+T$):NEXT S
60 DATA 7E,F6,60,D3,82,06,50
70 DATA 0E,81,0D,20,FD,05,20,F8
80 DATA E6,CF,D3,82,C9
90 DEFINT A-Z:PA=&H80:PB=&H81:PC=&H82:CW=&H83:OUT CW,&H90:OUT PC,0
100 DA=&HE000:PRINT CHR$(12)
110 LOCATE 5,0:PRINT "1  ERASE CHECK   2  WRITE"
120 LOCATE 5,2:PRINT "3  VERIFY        4 LOAD"
130 LOCATE 5,4:PRINT "5  MONITOR";CHR$(11);
140 INPUT "COMMAND";C
150 IF C>5 OR C<1 GOTO 140
160 ON C GOSUB 190,340,470,640,180
170 E=0:V=0:N=0:GOTO 140
180 MON:PRINT :RETURN
190 REM ** ERASE CHECK **
200 OUT 81,0:OUT CW,&H90
210 FOR AH=0 TO 7
220 FOR AL=0 TO 255
230 OUT PB,AL
240 OUT PC,AH OR &H80
250 DT=INP(PA)
260 IF DT<>255 THEN GOTO 270 ELSE GOTO 310
270 E=1:IF N=0 THEN LOCATE 0,5:WIDTH36,20:PRINT "ADDRESS","DATA";SPC(18);CHR$(11
);:N=1
280 A$=HEX$(AH*256+AL):D$=HEX$(DT):IF LEN(D$)=1 THEN D$="0"+D$
290 IF LEN(A$)=1 THEN A$="00"+A$ ELSE IF LEN(A$)=2 THEN A$="0"+A$
300 PRINT "  ";A$," ";D$;SPC(15);CHR$(13)
310 NEXT AL:NEXT AH:OUT PC,0:OUT PB,0:WIDTH 36,20
320 IF E<>0 THEN PRINT "DONT ERASE ROM" ELSE PRINT "ERASE CHECK OK !"
330 RETURN
340 REM ** ROM WRITE **
350 OUT CW,&H80:OUT 81,0:OUT PC,&H40
360 FOR AH=0 TO 7
370 FOR AL=0 TO 255
380 WD=PEEK(DA+AH*256+AL)
390 IF WD=255 GOTO 420
400 OUT PA,WD:OUT PB,AL
410 K=USR0(AH)
420 NEXT AL:NEXT AH
430 OUT PC,0:OUT PB,0:OUT CW,&H90
440 WIDTH 36,20:PRINT "DATA WRITE END"
450 GOSUB 470
460 RETURN
470 REM ** DATA VERIFY **
480 OUT CW,&H90:OUT 81,0
490 FOR AH=0 TO 7
500 FOR AL=0 TO 255:RA=AH*256+AL
510 OUT PB,AL
520 OUT PC,AH OR &H80
530 DT=INP(PA)
540 MD=PEEK(DA+RA)
550 IF DT<>MD THEN GOTO 560 ELSE GOTO 610
560 V=1:IF N=0 THEN LOCATE 0,5:WIDTH36,20:PRINT"ADDRESS      WRITE DATA      READ
DATA";CHR$(11);:N=1
570 A$=HEX$(RA):M$=HEX$(MD):D$=HEX$(DT)
580 IF LEN(A$)=1 THEN A$="00"+A$ ELSE IF LEN(A$)=2 THEN A$="0"+A$
590 IF LEN(M$)=1 THEN M$="0"+M$
595 IF LEN(D$)=1 THEN D$="0"+D$
600 PRINT "  ";A$;SPC(11);M$;SPC(12);D$
610 NEXT AL:NEXT AH:OUT PC,0:OUT PB,0:WIDTH 36,20
620 IF V<>0 THEN PRINT "LOAD DATA ERROR" ELSE PRINT "VERIFY OK !"
630 RETURN
640 REM ** DATA LOAD **
650 OUT CW,&H90:OUT 81,0
660 FOR AH=0 TO 7
670 FOR AL=0 TO 255
680 OUT PB,AL
690 OUT PC,AH OR &H80
700 DT=INP(PA)
710 POKE DA+AH*256+AL,DT
720 NEXT AL:NEXT AH:OUT PC,0:OUT PB,0
730 GOSUB 470
740 PRINT "LOAD END"
750 RETURN
```



と表示します．すべてFFHなら，

```
ERASE CHECK OK !
```

と表示します．

2．書き込み

E000HからE7FFHまでのデータをROMに書き込みま

リスト2　WAITサブルーチン

```
ORG     LD A,(HL)      E800   7E
        OR 60H           01   F6 60
        OUT 82H,A        03   D3 82
        LD B,50H         05   06 50
L1      LD C,81H         07   0E 81
L2      DEC C            09   0D
        JR NZ,L2         0A   20 FD
        DEC B            0C   05
        JR NZ,L1         0D   20 F8
        AND CFH          0F   E6 CF
        OUT 82H,A        11   D3 82
        RET              13   C9
```

す．また，書き込み後自動的にベリファイも行ないます．

3．ベリファイ

E000HからE7FFHのデータとROM内のデータを比較します．すべて一致していれば，

```
VERIFY OK!
```

と表示し，そうでなければそのアドレス，ライト・データ，リード・データを表示し，

```
LOAD DATA ERROR
```

を表示します．

4．ロード

ROM内のデータを，E000HからE7FFHに転送します．また，終了後ベリファイも行ないます．

これはROMのコピーに使えます．

5．モニタ

PC-8001のモニタをコールします．リターンするには，CTRL B（コントロールB）を入力します．

このコマンドはデータ・エリア（E000H～E7FFH）にデータを書き込むときに使います．

1～4のコマンドは実行中画面が消えます．これは，DMAをストップさせ実行時間を短くするためです（OUT81，0で消え，WIDTH36，20で現われる）．

■プログラム

特別な命令は使っていませんが，OR，OUT，INPなどが他のマイコンにないかも知れません．これは，マシン語のそれと同じです．この命令をBASICで実行できるからこそ，このプログラムがBASICで書けるのです．

しかし，書き込み時の50msのWAITだけは，うまくいかなかったのでマシン語で作りました(**リスト2**)．

また，書き込みのときデータが，FFHのときは，50msのWAITをせず次のアドレスに進みます．これは，少しでも実行時間を減らすためです．

PRINT文の前に少しゴチャゴチャしたところがありますが，これはエラーメッセージ用の物で，画面フォーマットを気にしたためこのようになってしまいました．なお，プログラムの中で使用している変数の一覧表を**表1**に示します．

変数一覧表

変数	意味
S	マシン語ストア用(リード文のデーターが入る)
PA, PB PC, CW	PPI8255のアドレス用
C	コマンドの入力用
E	イレーズ・チェック・ルーチン内でエラーがあったときE＝1になる．
V	ベリファイ ルーチン内でエラーがあったときV＝1になる．
N	イレーズ・チェック，ベリファイ・ルーチンで初めてエラーがあったとき，画面表示とラベルを表示する．N＝Dのとき初めてエラーがあったことになる．
AH	2716用アドレス上位用
AL	〃　〃　下位用
DT	2716内から読み出したデータ
MD	データ・エリアのデータ
A＄ M＄ D＄	エラーのとき，データ表示用に文字列にしている．

■使用感

1番長いライト・ルーチンでもベリファイを含め約3分です．ROMライタの名機，『ペッカー』にはかないませんが，かなり短いと思います．

書き込みテストは3社の物しか行なっていませんが，すべてエラーなく書き込まれています．

最後に

このPROMライタは，I/O'80年12月号の2パス・アセンブラで作成したプログラムをROMに書き込むために作ったもので，以下にその方法を示します．

❶OFFSETをE000Hとしてテキストを作成し，アセンブルする．

❷! コマンドでモニタへ行き，E900Hから21H，00H，80H，C3H，E5H，17Hと書き込む（Sコマンド）．GE900と入力すると16KのPCにイニシャライズされBASICに戻るので本プログラムをLOADする．

❸アセンブラに戻る必要があるときは，モニタにして，G8103とする．

*　　*　　*

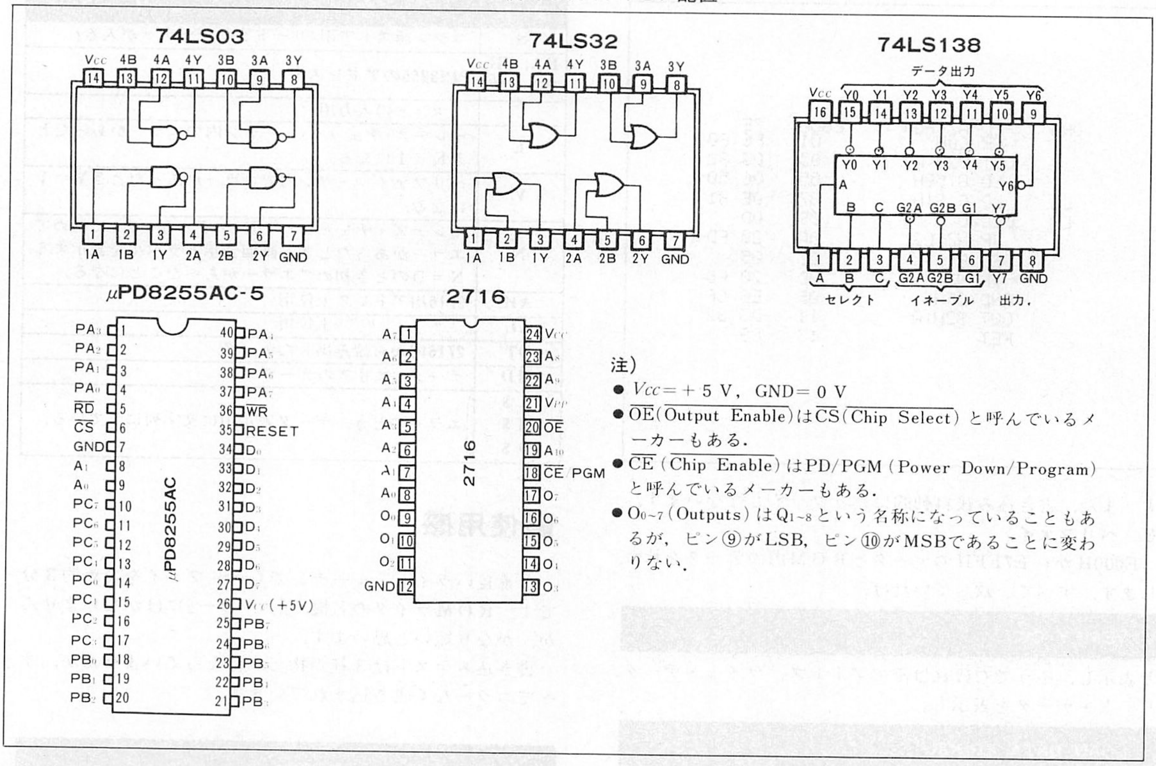
以上，いたらない説明でしたが，おわかりいただけたでしょうか．PCの回路が発表されましたが，まだまだハードウェアの記事は少ないようです．PCのハード派の皆さんガンバリましょう．

最後に使用したICのピン配置を図2に，2716のリード・ライト・タイミングを図3に示します．

参考文献

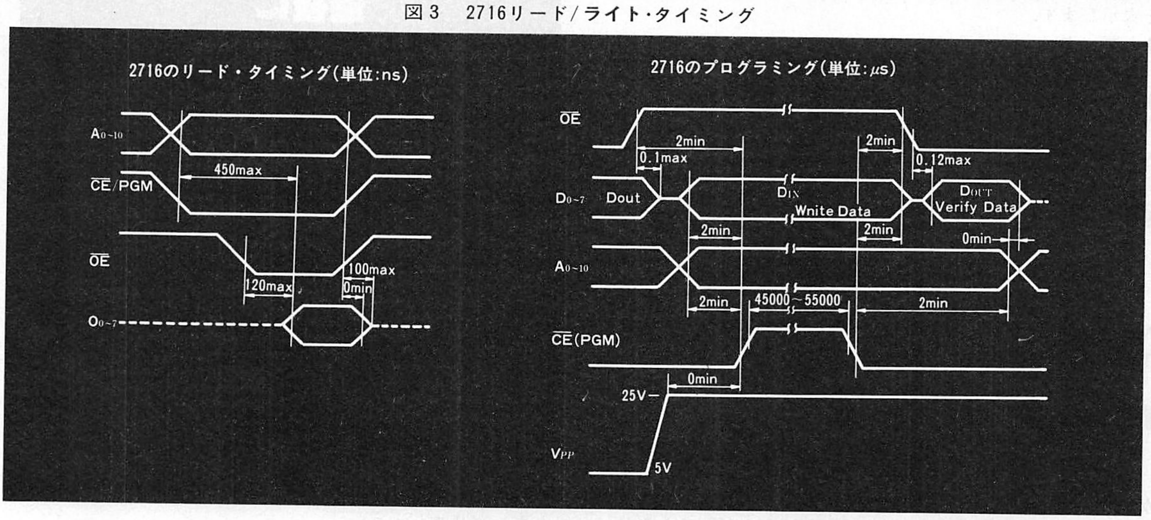
1)松本吉彦："続・わかるマイクロコンピュータ"8回，9回，トランジスタ技術，'80年5，6月号，CQ出版社
2)N-BASIC入門，アスキー出版
3)NEC μPD8255C マニュアル

図2　使用したICのピン配置



注)
- V_{CC}＝＋5 V，GND＝0 V
- $\overline{OE}$(Output Enable)は$\overline{CS}$(Chip Select)と呼んでいるメーカーもある．
- $\overline{CE}$(Chip Enable)はPD/PGM(Power Down/Program)と呼んでいるメーカーもある．
- $O_{0\sim7}$(Outputs)は$Q_{1\sim8}$という名称になっていることもあるが，ピン⑨がLSB，ピン⑩がMSBであることに変わりない．

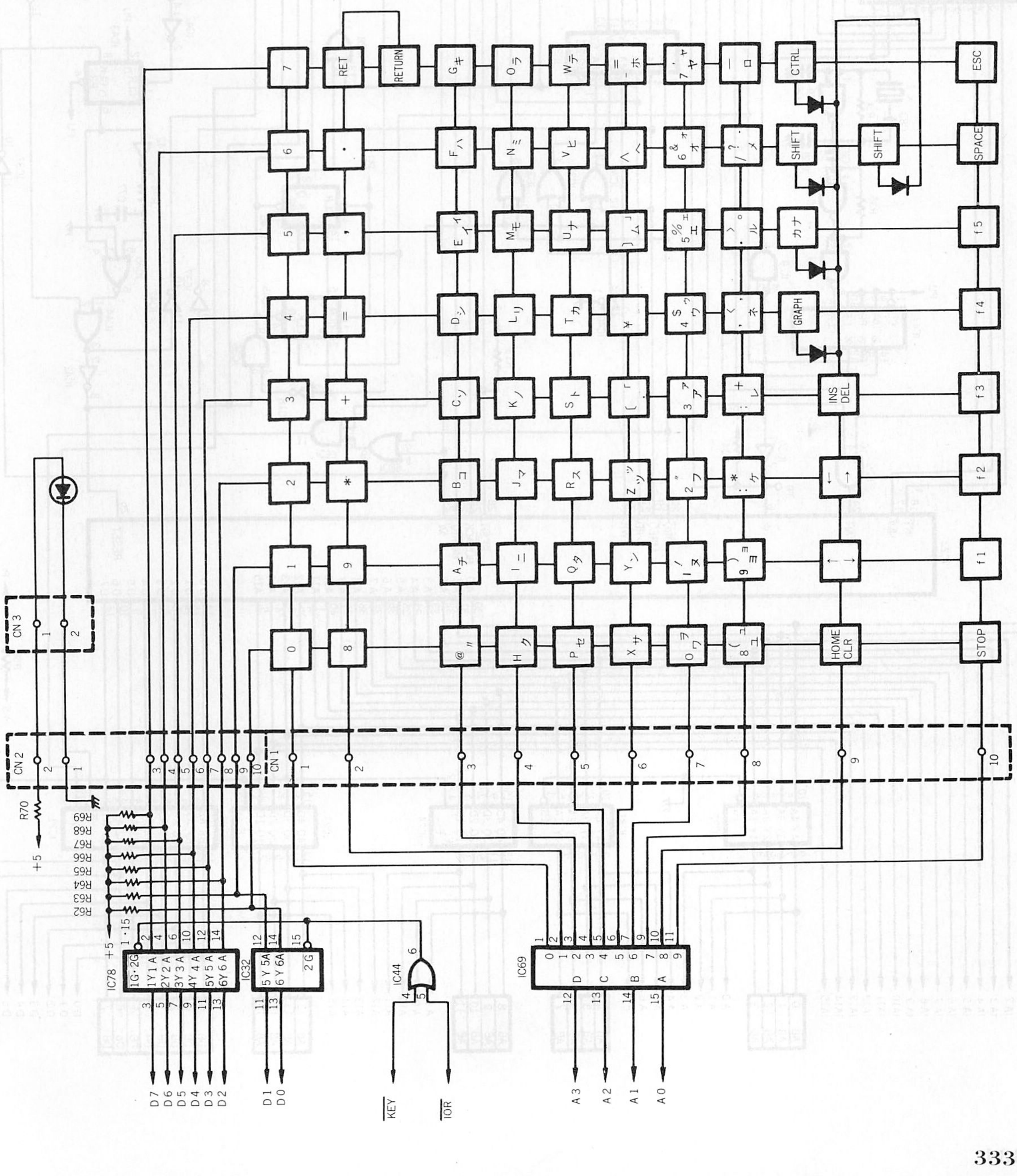
図3　2716リード/ライト・タイミング



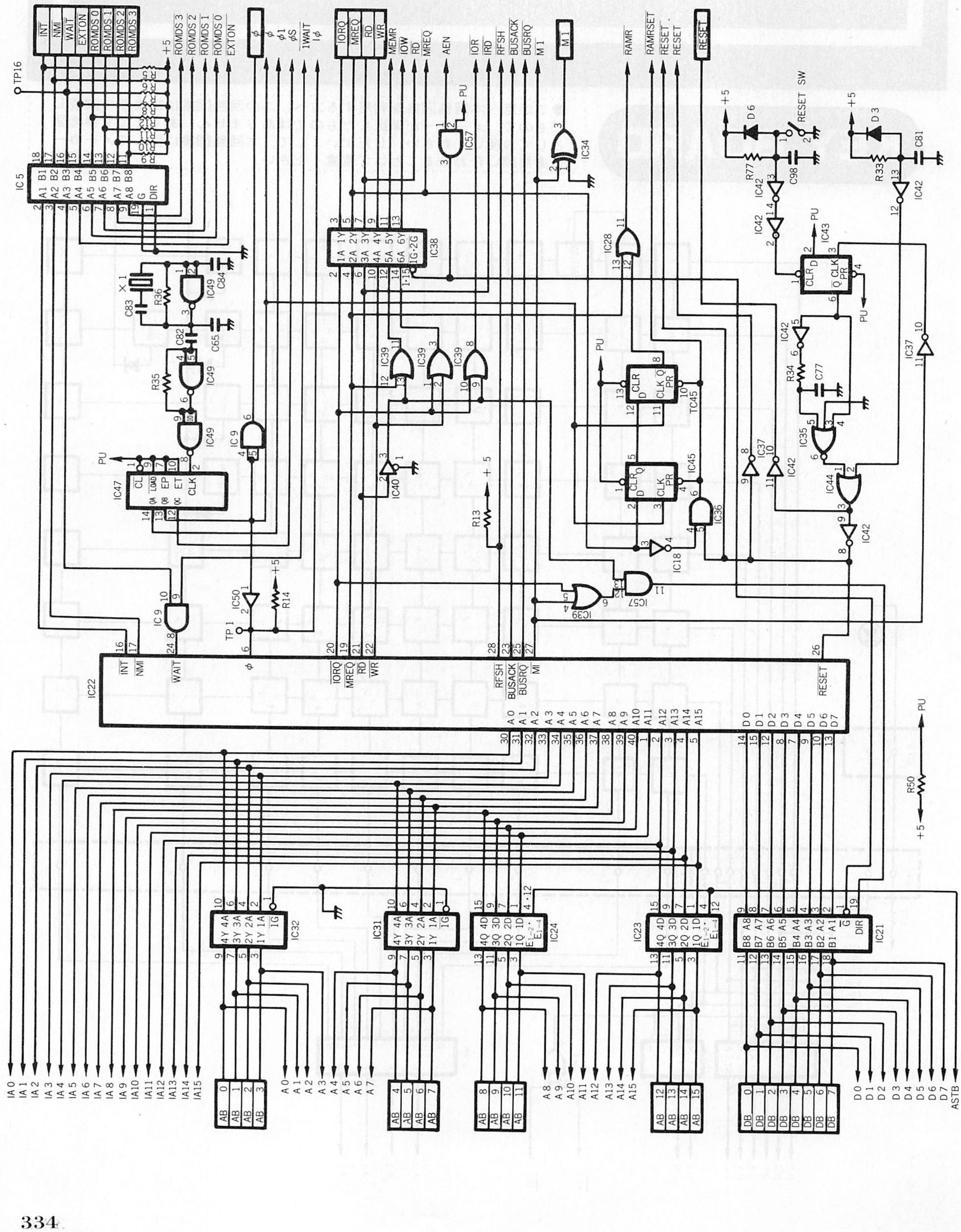
PC-8001全回路図集

KEY BOARD

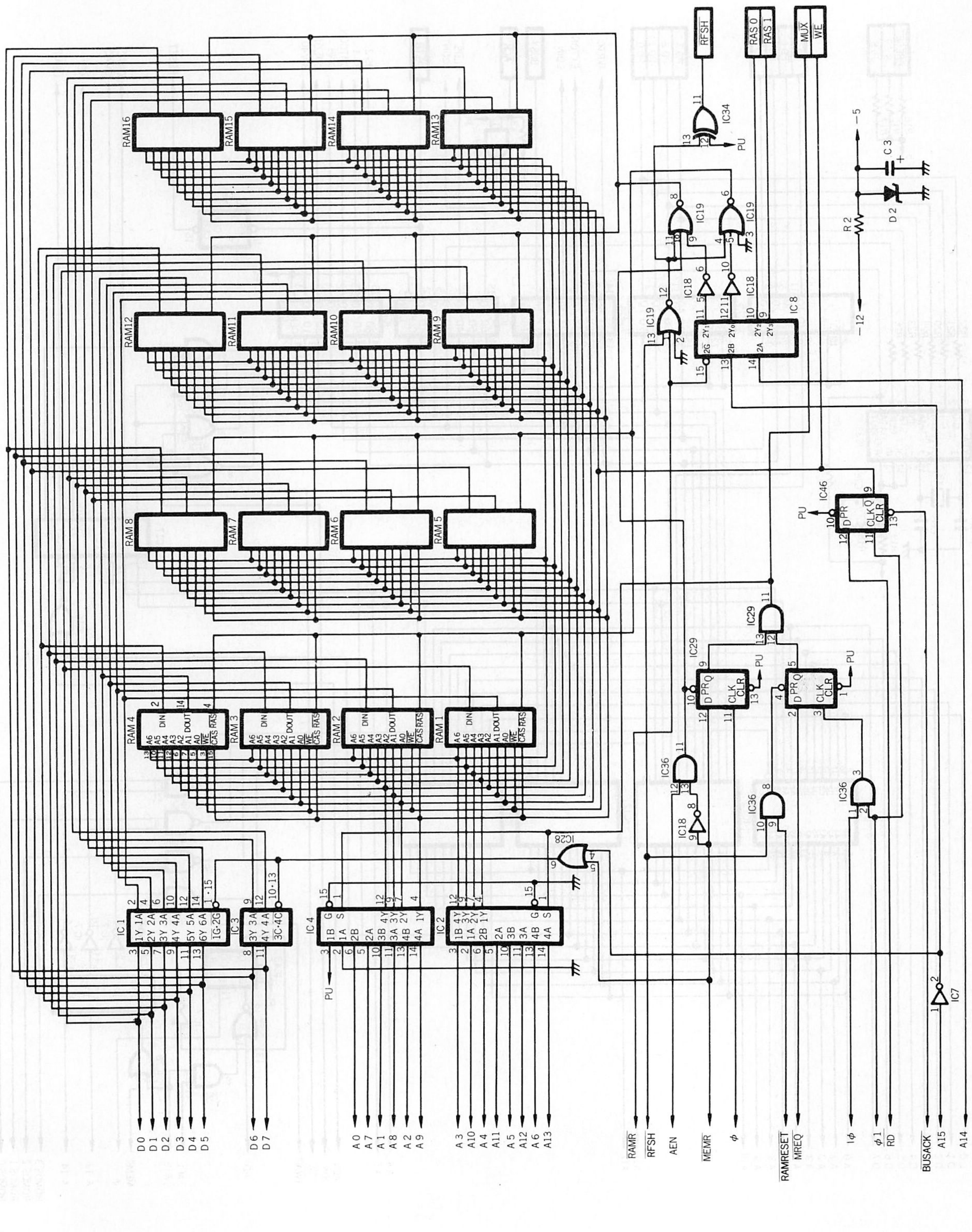
●ご注意（本回路図集を利用する方へ）この回路図集は編集部で調査したもので，メーカーが保証したものではありません．あくまでも参考図としてお使いください．したがいまして，本回路図集に関してメーカーに直接問い合わせることはご遠慮ください．（編）



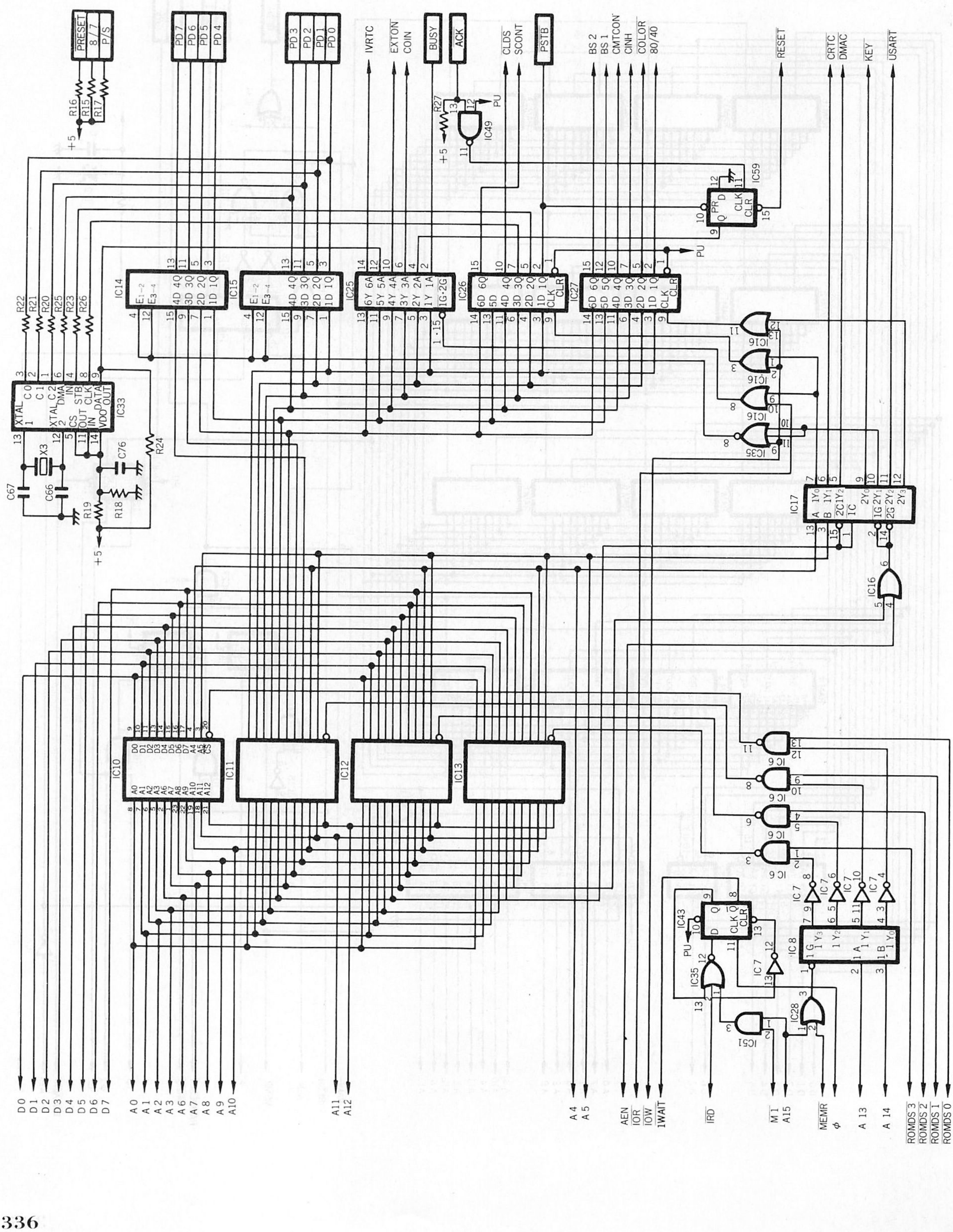
CPU


IC22
IC5
IC9
IC21
IC23
IC24
IC28
IC31
IC32
IC34
IC35
IC36
IC37
IC38
IC39
IC40
IC42
IC43
IC44
IC45
IC47
IC49
IC50
IC57
IC18
TP16
TP1
INT
NMI
WAIT
EXTON
ROMDS 0
ROMDS 1
ROMDS 2
ROMDS 3
IORQ
MREQ
RD
WR
M1
RESET
MEMR
IOW
AEN
IOR
IRD
RFSH
BUSACK
BUSRQ
RAMR
RAMRSET
RESET SW
ASTB
IA 0
IA 15
A 0
A 15
D 0
D 7
AB
DB
PU
+5

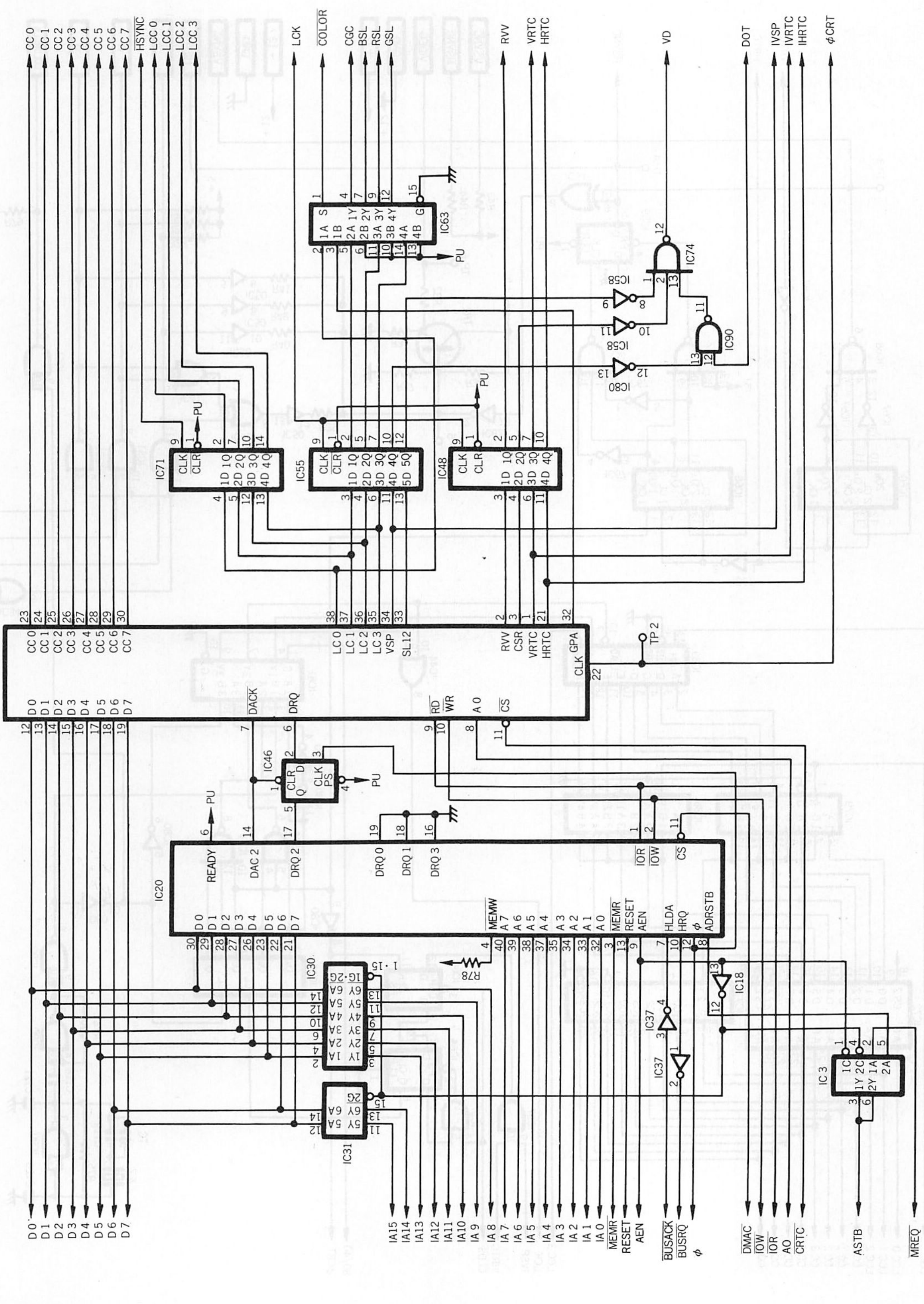
RAM


RFSH
RAS 0
RAS 1
MUX
WE
RAM16
RAM15
RAM14
RAM13
RAM12
RAM11
RAM10
RAM 9
RAM 8
RAM 7
RAM 6
RAM 5
RAM 4
RAM 3
RAM 2
RAM 1
IC 1
IC 3
IC 4
IC 2
IC 8
IC 7
IC18
IC19
IC28
IC29
IC34
IC36
IC46
PU
R 2
D 2
C 3
−5
−12
D 0
D 1
D 2
D 3
D 4
D 5
D 6
D 7
A 0
A 7
A 1
A 8
A 2
A 9
A 3
A10
A 4
A11
A 5
A12
A 6
A13
RAMR
RFSH
AEN
MEMR
φ
RAMRESET
MREQ
Iφ
φ1
RD
BUSACK
A15
A14

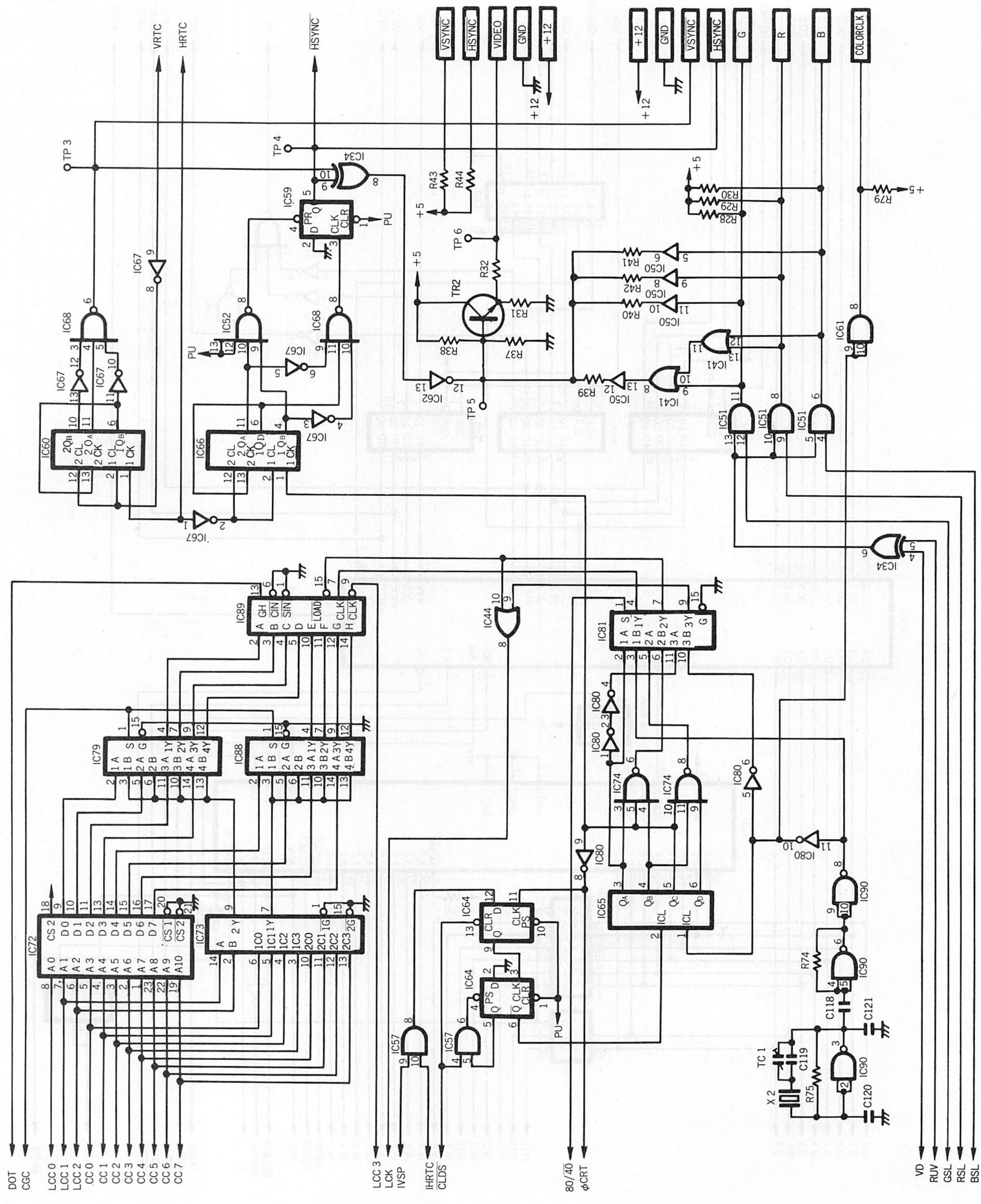
ROM & I/O


PRESET
8/7
P/S
PD 7
PD 6
PD 5
PD 4
PD 3
PD 2
PD 1
PD 0
IVRTC
EXTON
COIN
BUSY
ACK
CLDS
SCONT
PSTB
BS 2
BS 1
CMTCON
CINH
COLOR
80/40
RESET
CRTC
DMAC
KEY
USART
IC14
IC15
IC25
IC26
IC27
IC33
IC49
IC59
IC16
IC35
IC17
IC10
IC11
IC12
IC13
IC6
IC7
IC8
IC43
IC28
IC51
D 0
D 1
D 2
D 3
D 4
D 5
D 6
D 7
A 0
A 1
A 2
A 3
A 6
A 7
A 8
A 9
A 10
A 11
A 12
A 4
A 5
AEN
IOR
IOW
1WAIT
IRD
M 1
A15
MEMR
φ
A 13
A 14
ROMDS 3
ROMDS 2
ROMDS 1
ROMDS 0

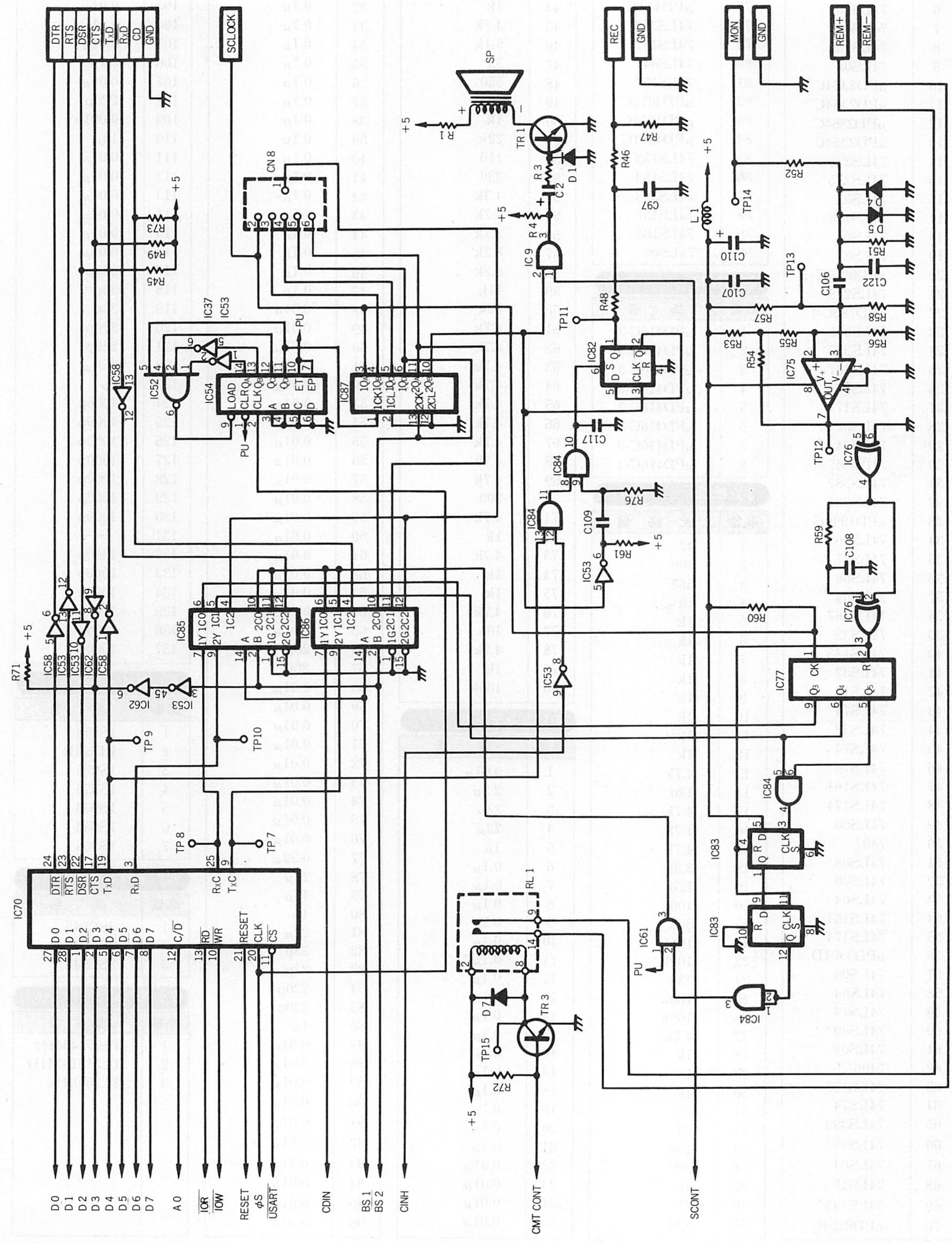
DMAC & CRTC


IC56
IC20
IC71
IC55
IC48
IC63
IC74
IC58
IC80
IC90
IC46
IC30
IC31
IC37
IC18
IC3
R78
TP 2
CC 0
CC 1
CC 2
CC 3
CC 4
CC 5
CC 6
CC 7
HSYNC
LCC 0
LCC 1
LCC 2
LCC 3
LCK
COLOR
CGC
BSL
RSL
GSL
RVV
VRTC
HRTC
VD
DOT
IVSP
IVRTC
IHRTC
φCRT
PU
D 0
D 1
D 2
D 3
D 4
D 5
D 6
D 7
IA15
IA14
IA13
IA12
IA11
IA10
IA 9
IA 8
IA 7
IA 6
IA 5
IA 4
IA 3
IA 2
IA 1
IA 0
MEMR
RESET
AEN
BUSACK
BUSRQ
φ
DMAC
IOW
IOR
AO
CRTC
ASTB
MREQ
READY
DAC 2
DRQ 2
DRQ 0
DRQ 1
DRQ 3
DACK
DRQ
RD
WR
A 0
CS
MEMW
HLDA
HRQ
ADRSTB
CLK
CLR
PS
GPA
CSR
VSP
SL12

VIDEO


VSYNC
HSYNC
VIDEO
GND
+12
G
R
B
COLORCLK
VRTC
HRTC
HSYNC
TP 3
TP 4
TP 5
TP 6
TR2
IC72
IC73
IC79
IC88
IC89
IC81
IC65
IC64
IC57
IC44
IC34
IC90
X 2
TC 1
DOT
CGC
LCC 0
LCC 1
LCC 2
CC 0
CC 1
CC 2
CC 3
CC 4
CC 5
CC 6
CC 7
LCC 3
LCK
IVSP
IHRTC
CLDS
80/40
φCRT
VD
RUV
GSL
RSL
BSL

CMT & SIO


DTR
RTS
DSR
CTS
TxD
RxD
CD
GND
SCLOCK
SP
REC
GND
MON
GND
REM+
REM−
CN 8
IC70
IC85
IC86
IC87
IC54
IC52
IC58
IC53
IC62
IC9
IC82
IC84
IC75
IC76
IC77
IC83
IC61
TR 1
TR 3
RL 1
L 1
TP 7
TP 8
TP 9
TP10
TP11
TP12
TP13
TP14
TP15
PU
+5
D 0
D 1
D 2
D 3
D 4
D 5
D 6
D 7
A 0
IOR
IOW
RESET
φS
USART
CDIN
BS 1
BS 2
CINH
CMT CONT
SCONT

PC-8001 パーツ・リスト

1）IC

品番	品　名
1	74LS367
2	74157
3	74LS125
4	74157
5	74LS245
6	74LS00
7	74LS04
8	74LS139
9	74LS08
10	μPD2364C
11	μPD2364C
12	μPD2364C
13	μPD2364C
14	74LS375
15	74LS375
16	74LS32
17	74LS155
18	74LS04
19	74LS27
20	μPD8257C-5
21	74LS245
22	μPD780C-1
23	74LS375
24	74LS375
25	74LS367
26	74LS174
27	74LS174
28	74LS32
29	74LS74
30	74LS367
31	74LS367
32	74LS367
33	μPD1990C
34	74LS86
35	74LS27
36	74LS08
37	74LS04
38	74LS367
39	74LS32
40	74LS125
41	74LS32
42	74LS14
43	74LS74
44	74LS32
45	74LS74
46	74LS74
47	74LS161
48	74LS174
49	74LS00
50	7407
51	74LS08
52	74LS20
53	74LS04
54	74LS161
55	74LS174
56	μPD3301D
57	74LS08
58	74LS04
59	74LS74
60	74LS393
61	74LS08
62	7406
63	74LS157
64	74LS74
65	74LS393
66	74LS393
67	74LS04
68	74LS10
69	74LS145
70	μPD8251C
71	74LS175
72	μPD2316EC
73	74LS153
74	74LS10
75	μPC271C
76	μPD4030C
77	μPD4024C
78	74LS367
79	74LS157
80	74LS04
81	74LS157
82	μPD4013C
83	μPD4013C
84	μPD4081C
85	74LS153
86	74LS153
87	74LS393
88	74LS157
89	74LS166
90	74LS00

2）RAM

品番	品　名
1	μPD416C-3
2	μPD416C-3
3	μPD416C-3
4	μPD416C-3
5	μPD416C-3
6	μPD416C-3
7	μPD416C-3
8	μPD416C-3

3）抵抗

品番	抵　抗　値
1	47
2	300
3	560
4	1k
5	1k
6	1k
7	1k
8	1k
9	1k
10	1k
11	1k
12	1k
13	4.7k
14	120
15	4.7k
16	4.7k
17	4.7k
18	3.3k
19	1.5k
20	100k
21	30k
22	100k
23	100k
24	33k
25	100k
26	100k
27	4.7k
28	1k
39	1k
30	1k
31	75
32	68
33	10k
34	100
35	1k
36	1k
37	430
38	240
39	200
40	160
41	820
42	390
43	1k
44	1k
45	4.7k
46	5.1k
47	33
48	550
49	4.7k
50	1k
51	22k
52	110
53	330
54	4.3k
55	2.2k
56	2.7k
57	8.2k
58	8.2k
59	11k
60	22k
61	4.7k
62	4.7k
63	4.7k
64	4.7k
65	4.7k
66	4.7k
67	4.7k
68	4.7k
69	4.7k
70	200
71	4.7k
72	1k
73	4.7k
74	1k
75	1k
76	4.7k
77	10k
78	4.7k
79	1k
80	100k

4）コンデンサ

品番	容　量
1	0.01μ
2	2.2μ
3	22μ
4	22μ
5	1μ
6	0.1μ
7	0.1μ
8	0.1μ
9	0.1μ
10	0.1μ
11	0.1μ
12	0.1μ
13	0.1μ
14	0.1μ
15	0.1μ
16	0.1μ
17	0.1μ
18	0.1μ
19	0.1μ
20	0.1μ
21	0.1μ
22	0.01μ
23	0.01μ
24	0.01μ
25	0.01μ
26	0.01μ
27	0.01μ
28	0.01μ
29	0.01μ
30	0.01μ
31	1μ
32	0.1μ
33	0.1μ
34	0.1μ
35	0.1μ
36	0.1μ
37	0.1μ
38	0.1μ
39	0.1μ
40	0.1μ
41	0.1μ
42	0.1μ
43	0.1μ
44	0.1μ
45	0.1μ
46	0.1μ
47	0.1μ
48	0.01μ
49	0.01μ
50	0.01μ
51	0.01μ
52	0.01μ
53	0.01μ
54	0.01μ
55	0.01μ
56	0.01μ
57	0.01μ
58	0.01μ
59	0.01μ
60	0.01μ
61	0.01μ
62	0.01μ
63	0.01μ
64	0.01μ
65	0.01μ
66	20p
67	20p
68	0.01μ
69	0.01μ
70	0.01μ
71	0.01μ
72	0.01μ
73	0.01μ
74	0.01μ
75	0.01μ
76	0.01μ
77	0.22μ
78	22μ
79	15μ
80	1μ
81	22μ
82	33p
83	33p
84	220p
85	220p
86	1μ
87	0.01μ
88	0.01μ
89	0.01μ
90	0.01μ
91	0.01μ
92	0.01μ
93	0.01μ
94	0.01μ
95	0.01μ
96	0.01μ
97	0.1μ
98	1μ
99	0.01μ
100	0.01μ
101	0.01μ
102	0.01μ
103	0.01μ
104	0.01μ
105	0.01μ
106	0.1μ
107	0.01μ
108	220p
109	0.001μ
110	10μ
111	0.01μ
112	0.01μ
113	0.01μ
114	0.01μ
115	0.01μ
116	0.01μ
117	220p
118	33p
119	33p
120	220p
121	220p
122	0.01μ
123	1μ
124	1000p
125	1000p
126	1000p
127	1000p
128	1000p
129	1000p
130	1000p
131	———
132	1000p
133	1000p
134	1000p
135	1000p
136	1000p
137	1000p

5）ダイオード

品番	品　名
1	1S953
2	RD5.1E
3	1S953
4	1S953
5	1S953
6	1S953
7	1S953

6）トランジスタ

品番	品　名
1	2SC945
2	2SC945
3	2SC945

7）X' tal

品番	品　名
1	15.9744MHz
2	14.31818MHz
3	32.768kHz

《ご注意》本回路図集はメーカーが保証したものではありませんので使用にあたってはご注意ください．（編）





I/Oの本

既刊・近刊

I/O別冊 ライブラリ・シリーズ

B5判 280頁 定価2,500円(〒300)

システム・プログラム①

買えば何十万円にもなる基本プログラムにアマチュアが挑戦！たちまち重版！

モニタ・エディタ・アセンブラ・逆アセンブラ・インタープリタ・コンパイラ……etc.

▶出てくるマシン…PC-8001・MZ-80・APPLEII・PET・M100……etc.

アプリケーション・プログラム①

マイコン用の価値ある応用プログラムを多数収録.

▶作表・2次元3次元表示・乱数・適性検査・成績管理・ワードプロセッサ・株式売買・家計簿・品質管理・合計処理・データベース……etc.

▶出てくるマシン…PC-8001・MZ-80・APPLEII・PET・M100……etc.

システム・プログラム②

No.1から1年，全国のマイコン・ファンの成果を再び.

I/O別冊

APPLE and PET B5判 280頁 定価2500円

APPLE/PETのユーザーと6502ファンのためのガイドブック

★APPLEディスク夜話 ★APPLE FORTH ★PET3032徹底研究 ★6502モニタ……etc.

プログラム電卓ゲーム A5判 180頁 定価1200円

『電卓コーナー』が本になった！

グラフィック・プリンタの使い方 B5判 212頁 定価1900円

MP-80ユーザー必携のハンドブック.

APPLE and PET② B5判 280頁 定価2500円

いよいよ第2弾が発行されます（7月下旬刊）.

I/O別冊『徹底研究シリーズ』

B5判 定価各1,900円(〒300)

別冊①『マイコン徹底研究』

M6800をハードからソフトまで初心者にもわかるように，ていねいに解説．マイコンの入門書として大好評！

別冊③『BASICゲーム徹底研究』

Tiny BASICやレベル1BASICのプログラミングの基礎から応用まで，徹底的に解説.

別冊④『マシン語徹底研究』

〝マシン語〟と聞いただけで〝ゾッ〟とするあなたのための入門書．Z80，8080，6800，6502を解説.

別冊⑨『マイコン・ゲーム徹底研究[2]』

HEAD-ON，スペース・インベーダー，Tiny与作など楽しいゲームを満載！

別冊⑩『マイコン・ソフト徹底研究』

アセンブラ入門からDOSの作り方まで，ソフトに強くなりたいあなたのための解説書.

別冊⑪『マイコン・ゲームの本[1]』

平安京エイリアンからグラフィック麻雀までPC-8001，MZ-80，ベーシックマスターL3のゲームを満載！

コンピュータ・ファン

No.4 ビジネス・ソフトの作り方

B5判 192頁 定価 1,900円(〒300)

給与計算・所得税計算・株式プログラム…etc.

I/O BOOKS

CAP-X入門〔56年度版〕

赤松 徹著 ¥1,900 (〒300)

たった12の命令を覚えるだけでアセンブラがわかる！ 情報処理技術者試験受験者ばかりでなく，アセンブラ入門者にとっても格好の入門書です.

PASCAL入門

マンチェスター大学 I.R.Wilson/A.M.Addyman著 ¥1,200 (〒250)

ＰＡＳＣＡＬを60もの豊富な例題でわかりやすく解説した本書は，ＰＡＳＣＡＬ入門書として全世界に愛読者を持ち，英・独・米・で出版されています．あなたも本書でPASCALをマスターしてください.

UCSD PASCAL演習

カリフォルニア大学Kenneth L.Bowles著 ¥2,900 (〒300)

あのUCSD PASCALの開発者Bowlesの著，〝Problem Solving Using PASCAL〟の翻訳が近々刊行されます．ご期待ください.

マイコンロボットの作り方

Tod Loofbourrow著 水島敏夫訳 ¥980 (〒250)

ロボットのフレーム作りから，マイコンによる制御のしかたまで徹底的にわかりやすく解説．アルミ材の加工の仕方，ICのピン接続，プログラム・リストなどが詳細に述べられています.

あなたもロボット『MIKE』を作ってみませんか？

対訳ポケット電卓ゲーム

Edwin Schlossberg/John Brockman著 ¥980 (〒250)

電卓で遊びながら英語をマスターしましょう！

著者はシュロスバーグ(科学・文学博士)とブロックマン（哲学）の名コンビ！

ディーラーをやっつけろ！

カリフォルニア大学エドワード・O・ソープ著 四六判 ¥1,800(〒250)

カリフォルニア大学の数学教授がコンピュータを使ってブラックジャックの必勝法を開発！おかげでラスベガスやプエルトリコのカジノは大恐慌．本書さえあれば，〝ミスターK〟も大負けしないで済んだ!?ギャンブラーのあなたの必読書！米国でベストセラー！

コンピュータ犯罪との戦い

アメリカ大学オーガスト・ベクエイ著 四六判 ¥1,200(〒250)

鉄道から貨車が200台蒸発．預金口座から数百万ドルが蒸発，コンピュータ犯罪は貧しい者が行なう犯罪ではない．その主役は若くて教養のある技術的に有能なエリートである.

ホワイト・カラー犯罪の中で大きな割合を占めるようになったコンピュータ犯罪を米国の第1人者が解説.

The Best of I/O

ザ・ベスト・オブ・アイオー

I/Oに掲載された主要記事を再編集しておとどけします.

定価各2,500円(〒300)

- No.1〔78年ハード編上〕
- No.2〔78年ハード編下〕
- No.3〔78年ソフト編〕
- No.4〔79年ハード編上〕
- No.5〔79年ハード編下〕
- No.6〔79年ソフト編上〕
- No.7〔79年ソフト編下〕
- No.8〔MZ-80活用研究（¥1900）〕
- No.9〔PC-8001活用研究〕

東京・代々木

工学社

〒151 東京都渋谷区代々木1-37-1

ぜんらくビル5F ☎(03) 375-5784㈹

振替口座 東京 5-22510

株式会社 工学社

編集後記

■PC-8001は発表当時，その価格と機能で大きな話題になりました．ワンボード・マイコン時代からパーソナル・コンピュータ時代への決定打だったと言っても過言でないくらいのインパクトでした．APPLEII, PET, TRS-80などの外国勢が日本に上陸してきたとき，現在のように国産のパーソナル・コンピュータが堂々とベストセラー・マシンになることを予測できた方が何人いたでしょうか．現在，PC-8001はホビーに，実務に多くのユーザーを獲得し，ソフトの数も海外のマシンと同等に揃いつつあります．'81年来にはPC-8000シリーズのファミリーとして，下位のPC-6000，上位のPC-8800シリーズも発表され，ますますユーザーの層が厚くなることが予想されます．

■本書に収録した記事は'80年に月刊誌I/Oに発表されたものが中心ですが，ユーザーがいかにPC-8001を料理していったか，その歩みがわかると思います．また，モニタの解析，アセンブラ，逆アセンブラなど，ソフト開発に必要な情報も満載しています．また，河合氏の『PC-8001入門』は，パーソナル・コンピュータ入門コースの講師としての豊富な経験を生かした，わかりやすい構成になっています．ビギナーの方はもちろん，かなりのベテランの方にも参考になる点が多いと思います．ぜひご一読を．

■なお，現在，毎月多くのＰＣに関する記事が掲載されており，『PC-8001活用研究』も続編を予定しています．

本書やI/Oをご覧になってＰＣに関する新しいアイデアが出たら，ぜひI/Oに投稿してくださるようお願いいたします．

I/O別冊

PC-8001活用研究

昭和57年 3月15日発行

定価2500円

編　集　I/O編集部

発行人　星 正明

発行所　株式会社工学社

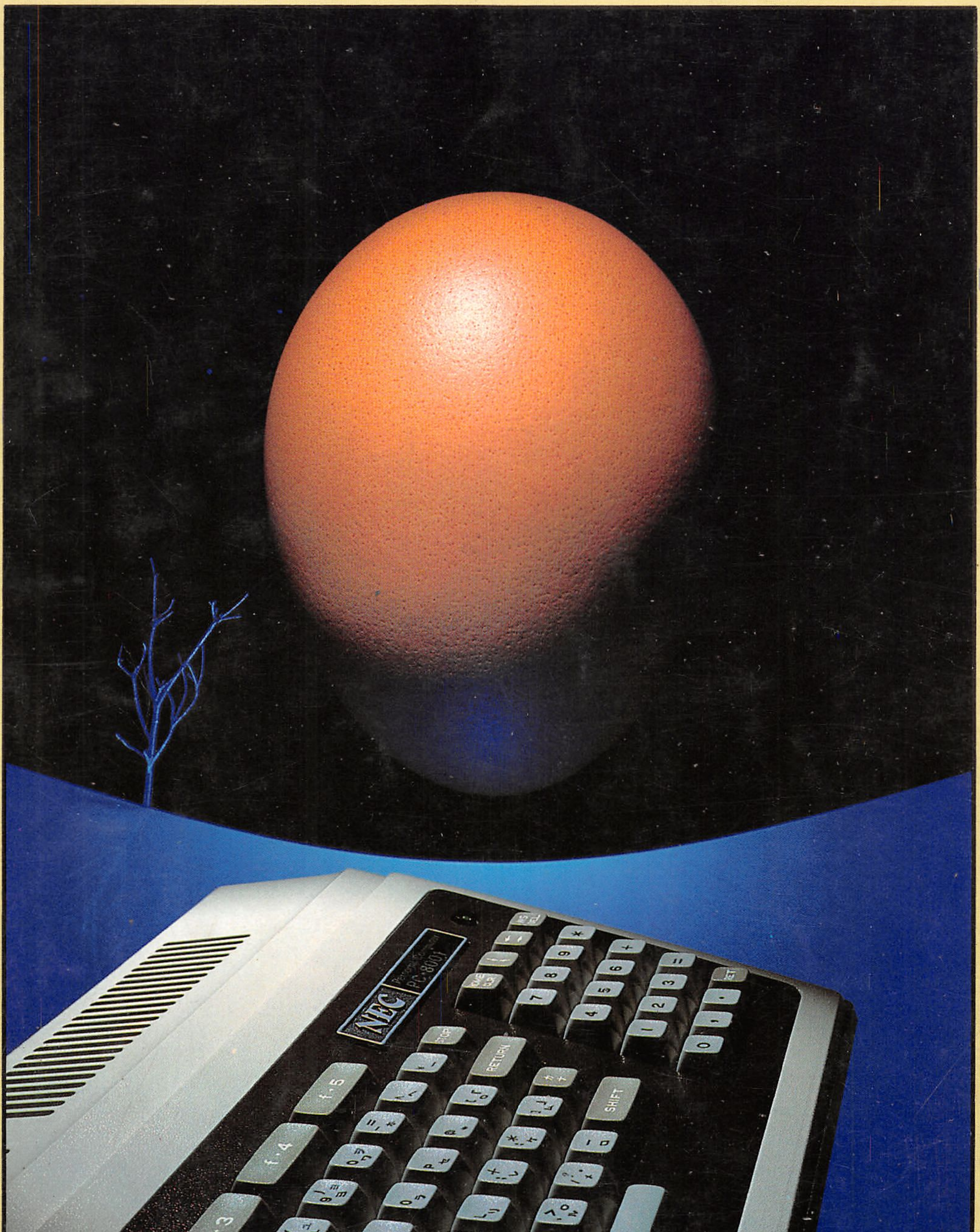
〒151 東京都渋谷区代々木1-37-1 ぜんらくビル５Ｆ

☎(03)375-5784㈹〔営業・広告〕

☎(03)320-1218㈹〔編集〕　振替　東京5-22510

印刷：㈱耕文社

PC-8001 活用研究



定価2500円

雑誌61473-44